• New Series For Beginners ➤ Introduction To Robotics

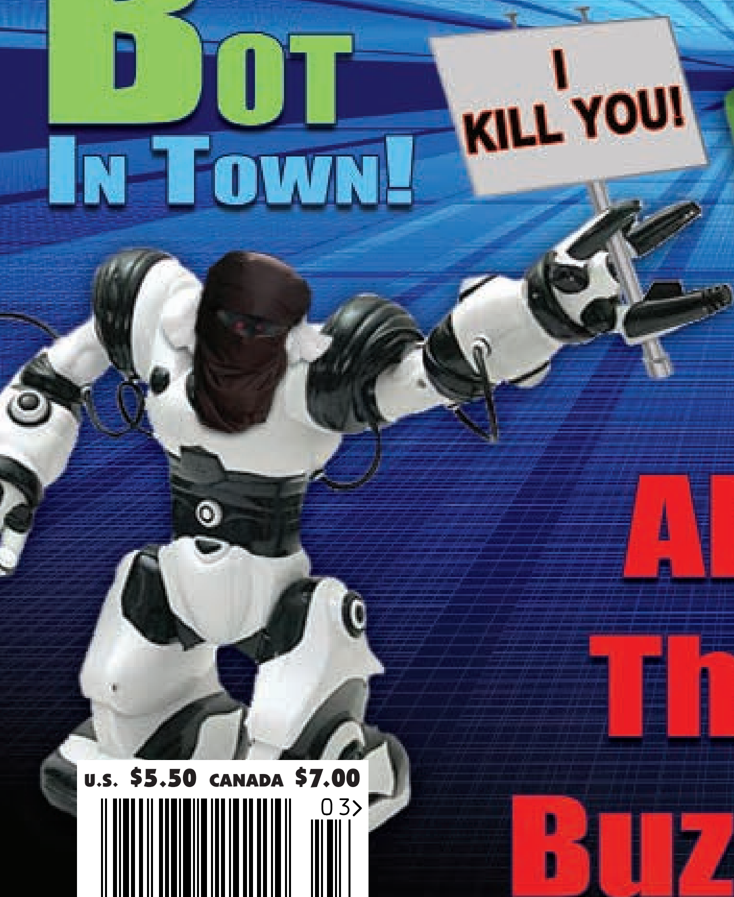
SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

March 2010

Move Over Robo — There's A New Bot In Town!



All The Buzz ...

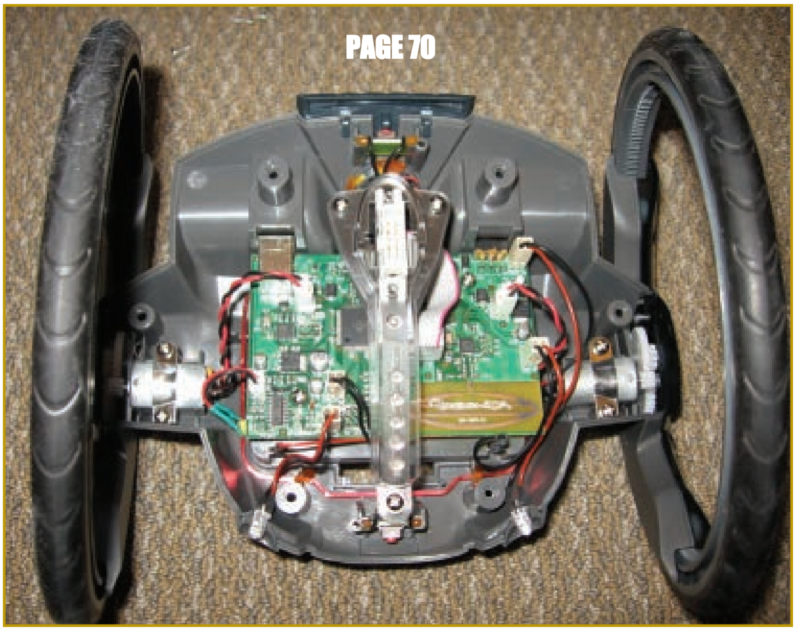
- VEX
- GPS
- COMBAT
- ROBOGAMES

U.S. $5.50 CANADA $7.00


The Evolution Continues
BIOLOID
PREMIUM Kit
New Release
Special Gift
Zigbee module
for 1,000 customers
All-in-one pacakge for
advanced robot builders
CE FC
BIOLOID
CM-510
Main Controller
RoboPlus
Gyro Sensor
Walk Balancing
DMS
Distance Measurement
Humanoid Skin
RC-100
Remote Controller
Li-Po Battery
11.1V 1000mA
USB2Dynamixel
Dynamixel AX-12+
18 DOF Actuators
IR Sensor
Object Detection
• Excellent humanoid walking performance (Adjusting posture while walking)
• Various sensors including Gyro, Distance, IR and more external ports
• Remote control capability (IR-default, Zigbee-optional)
• C-style programming & motion teaching with RoboPlus S/W (USB interface included)
• Transparent humanoid skin for customization
• Digital Packet communication with daisy chain topology
• Building various robots using versatile expansion mechanism
ROBOTIS
www.robotis.com
TEL : +82-70-8671-2600
Email : contactus2@robotis.com

SERVO MAGAZINE

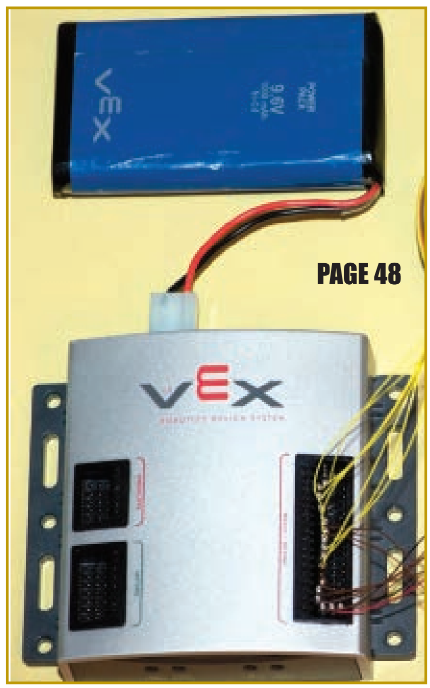
03.2010
VOL. 8 NO. 3

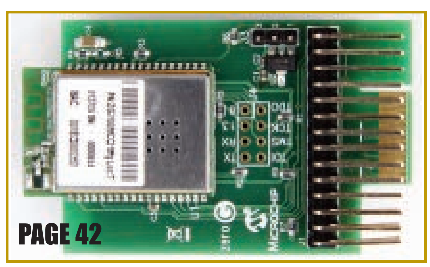


The Combat Zone...

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...





Mind / Iron

by Bryan Bergeron, Editor

Bots and Intelligent Agents

I'm writing this editorial with the assistance of a software robot or bot that I coded in AppleScript. This particular bot completes words, suggests sentence fragments, automates document formatting, and, in general, supercharges Word. The bot is autonomous in that it works in the background without my direct supervision. My goal, however, is to create an intelligent agent that can respond to my keyed or voice commands, such as "Write a summary paragraph based on information in Wikipedia/robots" and have the agent obediently extract the information from the web, construct sentences, and assemble the sentences into a coherent paragraph that matches the style and level of the other paragraphs in my editorial. No mean task.

The point in discussing bots and intelligent agents is that I believe they are key to the success of socially interactive mechanical robots. One obvious use for bots is in automating the learning process. Bots and intelligent agents can tirelessly comb through the Web in search of patterns in data, perfecting searches of the Web on their own. Wouldn't it be great if your robot could learn new words, phrases, and perhaps even objects by plugging into the Web?

Another application of this technology in robotics is communications. Chatter Bots, which have evolved significantly since Eliza was introduced in the 1960s, can form the basis of apparently intelligent dialogue between robots and humans. Research has shown that the elderly respond positively to the tactile and audible feedback from soft and furry mechanical pets, such as the PARO baby seal robot (**www.mahalo.com/paro-robot**). I strongly suspect that a social robot with more meaningful dialog would be even more appreciated by elderly users.

In addition to generating content for text-to-speech dialogues, Chatter Bots tied to video animation have been developed to exhibit appropriate facial expressions during voice recognition and speech synthesis. The benefit of this technology to mechanical robotics is obvious. Once you know the appropriate facial expression, displaying that expression on an animatronic face is relatively straightforward.

If you want to explore bots and intelligent agents, a good place to start is with one of the open source Chatter Bots. A relatively up-to-date website is AliceBot (**alicebot.blogspot.com**) which features an online Chatter Bot, as well as links to open source software. You might want to explore the rules for the Loebner Prize while you're at it (**www.loebner.net**). Each year, the Cambridge Center for Behavioral Studies awards $3K and a bronze medal for the most human-like computer. If you want the grand prize — a gold medal and $100K — your Chatter Bot will have to pass the Touring test. By the way, if you manage to create a bot that's indistinguishable from a human, you can count on space in this magazine for an interview.

Back to my relatively trivial AppleScript Bots ... you can also gain insight into various means of training or programming new behaviors in a mechanical robot by working with simple scripting languages. If you use the Unix OS, then you're probably intimately familiar with dozens of bots available for automating repetitive tasks. As you'll discover, scripting is good when you have a known endpoint and fixed process to get there. Problems arise in robotics when decisions aren't clear cut, when there are no specific rules that apply to a novel situation, and when the process becomes non-deterministic. You may ultimately have to rely on genetic algorithms, neural networks, and various Bayesian techniques to create something that can begin to pass the Touring test. However, you can use simple scripting and script generators to test your algorithms and overall approach to adding interactive intelligence to your next robot project. SV

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, David Geer, Dennis Clark, R. Steven Rainwater, Fred Eady, Kevin Berry, Chris Savage, Daniel Ramirez, Brennon Williams, Marco Meggiolaro, J. Miles, Pete Smith, Nick Martin, Ray Billings, Evan Ackerman, James Baker, Bryce Woolley, Evan Woolley

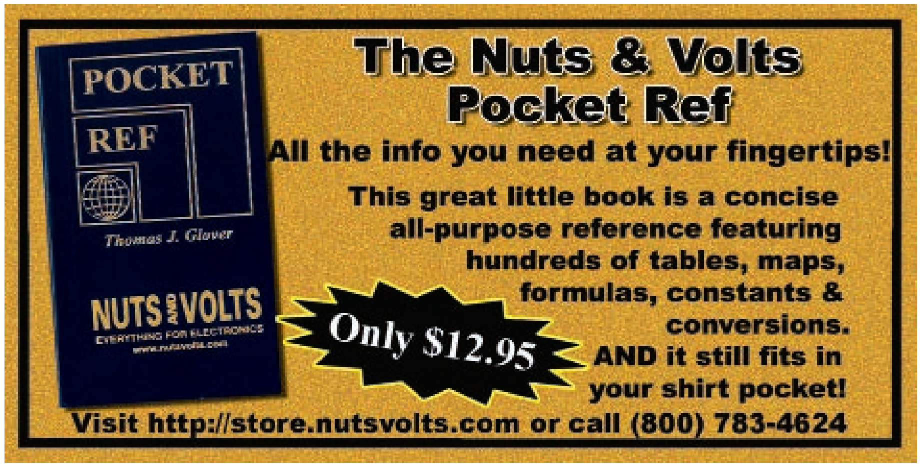
CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

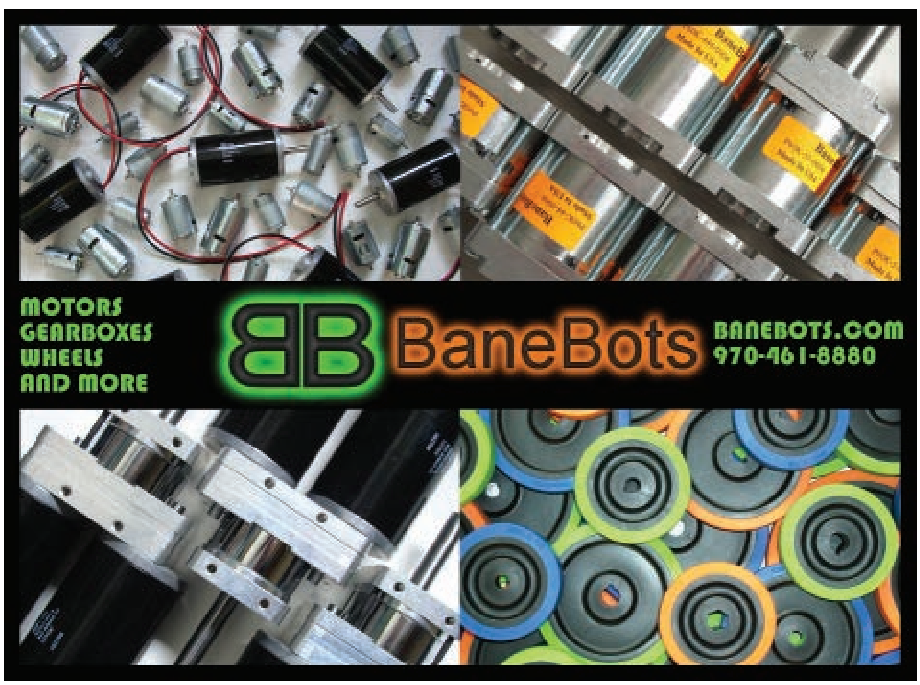
MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

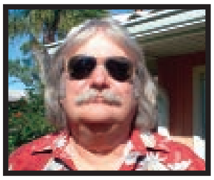
WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen







Robytes

by Jeff and Jenn Eckert

Ford's Touchy-Feely RUTH

Ford's RUTH robot passes judgment on a car interior. Courtesy of Ford Motor Co.

Upon hearing that Ford Motor Co. (**www.ford.com**) is working with a robot named RUTH, one might jump to the conclusion that a sort of mobile Roxxxy might be part of a future option package — the Charlie Sheen edition, perhaps. Fortunately, it's nothing that creepy. The name derives from Robotized Unit for Tactility and Haptics which is not quite tortured enough to win a coveted Bloated Labyrinthine Acronyms for Hype award ... but close. RUTH is also not particularly feminine. She is good at twiddling knobs, flipping switches, and adjusting air vents, however. It turns out that she has been helping engineers at Ford's European Advance Research Centre in Aachen, Germany to develop the interiors for almost all current and future Ford models.

In olden days, designers would actually put humans behind the wheel to determine which controls and dial settings were pleasing to most people. Human preferences have now been stored as digital data, and RUTH sits in the driver's seat and decides if she's comfy or not by measuring things like friction, roughness, softness, and temperature, and comparing these measurements with the consumer feedback database. So, while she does help make you comfortable, there's nothing personal about it. Sorry, Charlie.

Bot Performs Reverse Vasectomies

If you do feel the urge to merge your private parts with the world of robotics, consider a new procedure developed at the University of Florida (**www.ufl.edu**) that allows urologists to, um, cut off about 20 minutes of surgery time when performing a vasectomy reversal. And it's not just a matter of, um, shaving off a few minutes. At least in the short term, the robotic procedure results in higher sperm counts (54 million vs. 11 million, if anyone's counting). In addition, surgeons who perform reverse vasectomies typically use microscopes to watch what they're doing, and the new technique provides "less discomfort for some surgeons who would otherwise stand or sit with their backs bent for extended periods over a microscope."

Sijo Parekattil, M.D., demonstrates robot-assisted surgery. Photo by Sarah Kiewel/ University of Florida.

If discomfort translates into slips of the scalpel, by all means let's make sure the doc is deliriously comfortable. One drawback is cost, as the robotic procedure can run you $3,000 more than a standard one. But after all, your car isn't the only thing you really want to be in good hands.

Another Eye in the Sky

Corporate and military bigwigs at the Heron delivery ceremony.

We hear a great deal about heavily armed UAVs, e.g., the Predator, operating in Afghanistan, but quite a few are built for missions other than combat. Such is the Heron system, built by Israel Aerospace Industries (Royal Australian Air Force) and recently adopted by the Royal Australian Air Force (**www.airforce.gov.au**) for intelligence, reconnaissance, and surveillance (ISR) missions. The system includes an aircraft, mission payloads, a ground control station, spare parts, and ground support equipment. The Heron itself is a medium altitude long endurance (MALE) bird that can remain airborne for 30+ hours, cruise at 30,000 ft, and carry about 250 kg (550 lb) of payload. Cruise speed wasn't specified, but it has an operational range of "several hundred kilometers." Other specs include a 16.6 m (54.5 ft) wingspan and a takeoff weight of 1,200 kg (2,645 lb). The system is slated to operate for a year, with an optional extension for two more years.

"Perch and Stare" UAV

Cyber Technology's CyberQuad, a UAV with only four moving parts.

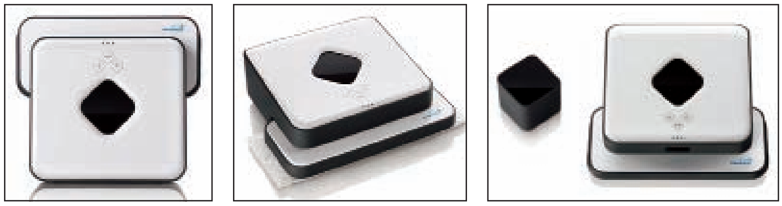
Another interesting UAV is the CyberQuad, offered by an Aussie company called Cyber Technology (**www.cybertechuav.com.au**). What you get is an electric vertical takeoff and landing (VTOL) platform that operates with only four moving parts. According to the company, "CyberQuad is a unique amalgamation of state-of-the-art VTOL UAV technologies, combining the mechanical simplicity, low noise, stability, and agility of a quadrotor with the compactness, safety, and efficiency of ducted fans."

The little nubbin attached to the frame is a camera which may tip you off that CyberQuad's main mission is to "perch and stare" at whatever is below it. But "stare" doesn't necessarily mean visual. It could be fitted with other types of sensors including, for example, an acoustic device that detects gunfire on a battlefield, determines its location, and relays coordinates back to the good guys. It can also carry sensors to detect pollutants or chemical warfare agents. In a recent demonstration, the little chopper conducted a survey of an offshore drilling platform and an oil rig that had been damaged by fire, flying around and inside the structures to provide a close look at the situation.

Top speed is about 44 mph (70 kph) with a climb rate of 33 ft/s (10 m/s), and it can hover for up to 40 min, depending on the mission. No price was specified, but if the thing's cheap enough for the consumer market, one can't help but marvel at the possibilities for mischief.

Make Your Floors Minty Fresh

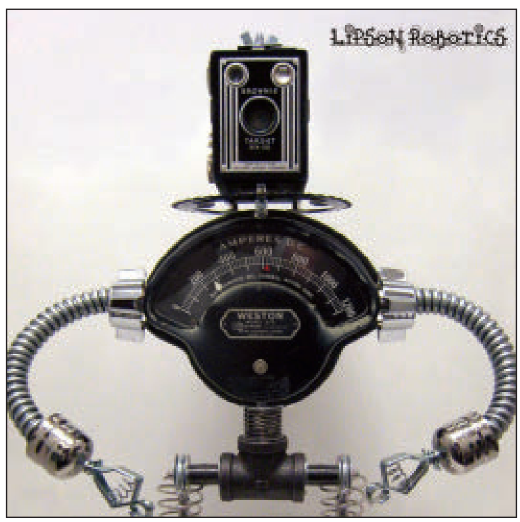
Okay, it's just another floor-cleaning bot. But if you've already traded in the dust mop for a Swiffer® and decided that it's just another back-breaking mop after all, you can automate the swiffing process for about $250. The Mint — a development of Evolution Robotics (**www.evolution.com**) — uses Evolution's NorthStar® technology and three different on-board positioning systems to map the room, plot out areas for cleaning, and pick up spots that it misses on the first pass. It employs a special mopping motion and Swiffer cleaning pads to remove dust, dirt, and hair from hard-surface floors automatically. Sensors keep it from moving onto rugs and carpets, and it adjusts pad friction for optimal traction on slippery floors. Mint is not slated to be in retail outlets until the fourth quarter of the year, but you can sign up to receive information about availability and purchase options by visiting **www.mintcleaner.com**.



Evolution's Mint floorbot uses Swiffer pads for cleaning.

Custom Robot Art Offered

This month's entry in the nonfunctional robot category is a series of fauxbots built by New Yorker David Lipson (**www.lipsonrobotics.com**) out of miscellaneous pieces of junk. He has built more than 150 of them over the years, and some are available for sale at **www.etsy.com**. My personal favorite is Weston, a 25 in tall piece that includes a Brownie camera and an old Weston meter. It is offered for $799, which unfortunately is roughly 100 times what I'd be willing to shell out. The meter doesn't work, anyway. Some are less than $300 which ain't bad for an original, unique sculpture. **SV**



Weston, a creation of sculptor David Lipson.

GEERHEAD

by David Geer
Contact the author at geercom@windstream.net

The Ultimate Buzz Lightyear Programmable Robot

With seven motors, a microprocessor, and multiple sensors, your favorite Space Ranger is no slouch among modern, commercially available robots.

Ultimate Buzz Lightyear walks forward, backwards, turns left and right, shoots an arm laser, talks to his arm communicator, has over 100 sayings in the original Tim Allen voice, and turns his head and blinks his eyes. What more could you want in an action figure? Well, quite honestly, Buzz has a whole lot more goin' on.

The variety of speech and motion options he has seems to go to infinity and beyond! (Okay, maybe not beyond infinity, but he is packed with lots of impressive features.) Buzz can "give you 5," speaks with synchronized lip movement, has multi-direction motion detection, obstacle avoidance, and an autonomous roaming mode. These are just some of the highlights sure to whet any roboticist's appetite for hacking!

Just like in the movies, Buzz believes he's a real Space Ranger that has crash-landed on a strange planet. If you tell him he's a toy, his behavior changes. The "Space Ranger" and "Toy" modes each offer their own unique set of phrases.

There is also an interactive target game in which you engage in an intergalactic battle with Buzz using the included infrared remote control. Buzz will be sure to tell you if your shot is a hit or miss.

Seven powerful motors take care of the eye, mouth, head, arm, waist, and leg movements. There are 15 command buttons for the original voice, actions, sound effects, animation, as well as other aspects of the robot.

Buzz was a winner of the 2009 Honors Award from the NAPPA (National Parenting Publications Awards). He stands 16" tall and you'll need about 14 batteries for full-function play.

Let's go through some of the different options a little more closely, to really see what the "buzz" is all about.

Programming and Walking

Buzz is endowed with a programmable puppeteer feature that records animated movements and poses that you create. Press the Programming button (the red button below the Lightyear chest emblem) and listen for the confirmation sound. Buzz will stand at attention with his arms at his sides. This is where you then move Buzz into the desired stance you want and press the Programming button again. You'll hear a dual beep.

Next, you'll press the circular arrow key on the included remote and Buzz will move using the programmed gesture. You can also program a series of movements by pressing the Programming button after each position you place Buzz in. You can enter up to 64 movements, gestures, sound effects or even include the laser light.

To make Mr. Lightyear walk, you'll use the remote control to send commands via infrared which Buzz receives via his front and rear IR sensors. The arrow keys on the remote control enable Buzz to walk in whatever direction you select.

I tried to get Buzz to walk on rugs and carpeting, but he failed to make any real headway. Instead, his legs remained stationary and the force of attempting to walk caused his upper body to gyrate. He did eventually cover about a foot of ground on a carpeted surface, but it took him a lot longer than it did to traverse a smooth surface.

Voice Command Mode

After you repeat one of six preset voice commands (in either Space Ranger, Toy Buzz, or Talk Back mode), Buzz will respond with one of 100 available verbal phrases in the popular Tim Allen voice. In a first-hand test, Buzz responded to various random comments with replies ranging from a simple "yes" to "Do you people still use fossil fuels or have you discovered crystallic fusion?" There are many other monologues about the gamma quadrant, various sectors, and an emperor who happens to be a sworn enemy of the galactic alliance. Several of the vocal responses are unprompted storytelling about the robot's mission.

Buzz's speech recognition technology accurately identified the pre-trained phrases whether they were spoken with different tones, emphasis, or inflections.

To program voice options, you press either the Space Ranger or Toy Buzz and Programming buttons simultaneously. Reprogramming the robot or turning it off erases all previous user programming.

Interactive Target Game Play

The Interactive Target Game makes for a fun but friendly battle. Once in game mode (press the remote key that looks like a gun site), the player aims the remote at the robot's waist and presses the site key to fire on the moving target which is, of course, Buzz Lightyear.

It appeared to me that any shots made at

Here, you see Buzz with his blue Space Ranger Mode button, green Microphone on-off button, and red Toy Buzz Mode button clearly visible on his chest. The round, red button at the right is the Programming button for entering and exiting Programming mode. You can also see the remote control.

A small Philips screw holds the center battery cover on the robot's back in place. Beneath that, sits three required AAA alkaline batteries. Buzz also requires four AA alkaline batteries in each leg to walk.

Here a mysterious hand presses the Buzz Lightyear laser button, lighting up the forearm "laser" LED.

The remote control with all 15 keys shown.

Here's the infrared transmitter.

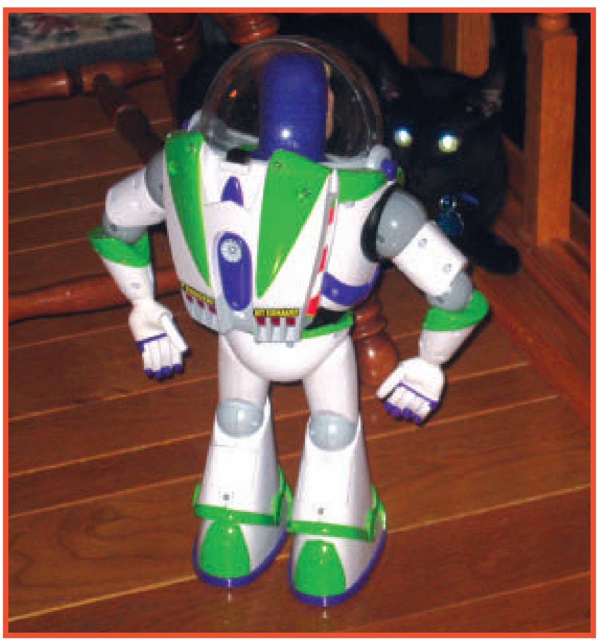
Buzz while he was talking did not register. Buzz will break up the game play now and then with brief comments. However, when Buzz is not speaking, he consistently registers hits and misses, and reports them to the player.

Remote Control Action

Directional buttons on the remote turn Buzz left and right, and move him backward and forward. Function and Action Command buttons activate Buzz in one of 11 ways.

For example, in Space Ranger (SR) mode, the robot responds with sayings and actions based on the assumption Buzz is truly a Space Ranger. In Toy Buzz (Buzz) mode, Buzz responds with his newfound understanding that he really is a toy.

Our cat doesn't quite know what to make of Buzz which is about how she reacted to the robot mouse she got for Christmas.



The microphone button turns the built-in mic on and off. Action buttons include Fire Laser, Star Command, Salute, Gimme 5, and To Infinity and Beyond — each prompting a vocal and physical sequence such as laser firing, calling star command, and giving a high five to the user.

Conclusion

In my opinion, the Ultimate Buzz Lightyear robot is closer in appearance and action to the "real deal" than any other replica or toy has come. Especially when his head and eyelids move. While some data on the Thinkway Toys site suggests that his wings pop out and light up, this did not happen when I tested the product.

Children of all ages can turn a passion for learning into hours of fun with Buzz, whether he's played with straight out of the box, or gets a few refinements from his "superior officer." **SV**

Resources

Ultimate Buzz Lightyear toy maker
www.thinkwaytoys.com

Explore Ultimate Buzz Lightyear videos on Google and YouTube.

PCmag video on YouTube
http://bit.ly/18wvg6

Engadget video on YouTube
http://bit.ly/VkWUm

Engadget article about Buzz
www.engadget.com/2009/06/26/ultimate-buzz-lightyear-robot-to-the-rescue-on-video

KidsTechReview article
www.kidstechreview.com/2009/12/review-ultimate-buzz-lightyear-robot

Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by
Dennis Clark

Well, the Christmas season has come and gone, and it appears that Santa was good to many of you robot builders out there since I've been getting some interesting questions. So, let's get started!

Q. Regarding the October 2008 Zombie.zip article on the R/C receiver decoder. I found the referenced implementation of a "tank-like" RC decoder very elegant and would like to try it out myself.

1) Please elaborate on the minimum support PIC hardware that I will need to implement your R/C decoder solution (i.e., in detail what development platform was used for the PIC 18F252).
2) Please elaborate on where I can obtain the support software (i.e., CCS PCH compiler).
3) Can this solution be ported to the PIC 16 series or is the interrupt structure that you used the reason you chose to upgrade to the PIC 18 series of processors?
4) Are there any other articles or reference information that will assist me in testing out this solution? Many thanks.

— Stephen

A. That project was very minimal and I also felt elegant; thank you for noticing.

I used the 18F252 just because it is a popular, all-around useful mid-range PIC. This project is pretty small code-wise, so one could go as small as the 18F2220 and have everything that I like in a chip — namely two PWM channels and a USART for debugging. This part is a little cheaper than the 18F252, as well.

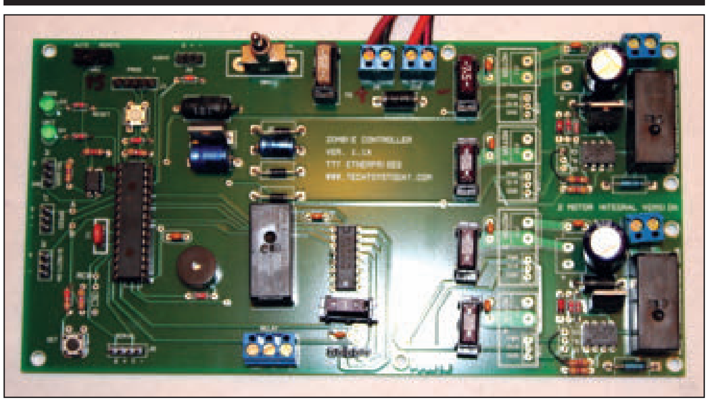
You can get the CCS PCH compiler from Custom Computer Services, Inc.; their website is **www.ccsinfo.com**. They have a variety of reasonably priced compilers for the various PIC microcontrollers. PCH will integrate into the Microchip MPLAB IDE seamlessly.

There aren't all that many 16 series PICs that have two PWM channels, and they aren't that much cheaper than the 18 series parts. Also, the 18 series is about twice as fast and the interrupt system is much more convenient. The 18F parts have a single-deep fast interrupt that saves the W and STATUS register when using the High Priority IRQ (Interrupt Request.) If you are careful with your coding and know exactly what you are modifying, then you don't need to do any housekeeping in your ISR (Interrupt Service Routine). The zombie.zip source code shows just such a lean ISR.

As far as other reading material on this topic, try hooking up with other builders on forums.

Let me expand a bit on the concept of a radio remote controlled platform. I used a common 75 MHz surface band radio system. If you want to build an RC based one, then I recommend also using a 75 MHz, or better yet, one of the new 2.4 GHz band radios. I used a two-channel, two stick

Figure 1. PIC18F252 controller.



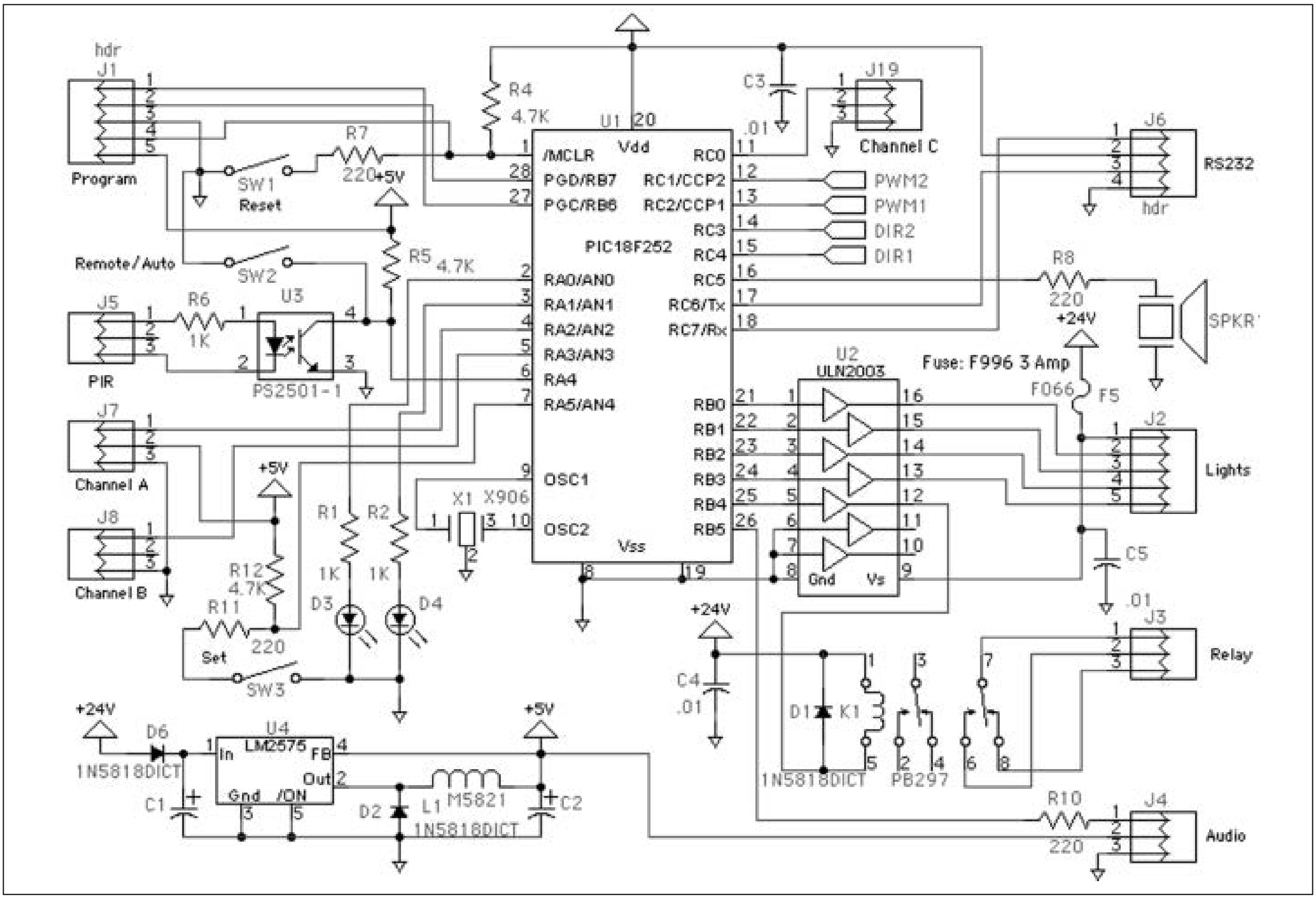
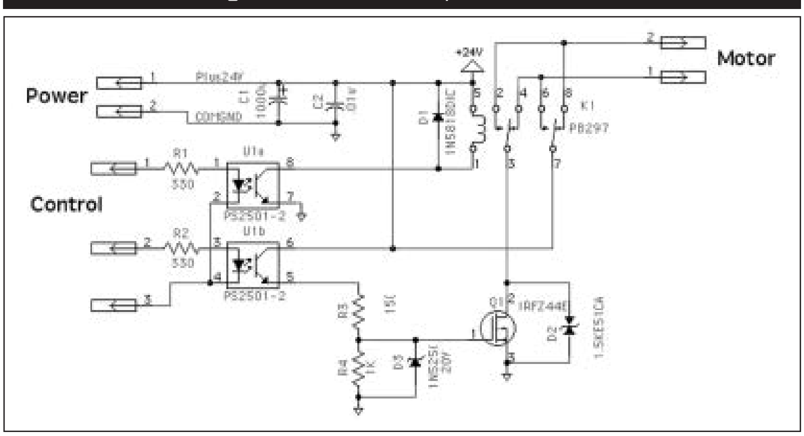


Figure 2. Controller schematic.

transmitter rather than a pistol grip style because it was easier to implement. I took my transmitter apart and rotated the left stick so that both sticks moved up and down the same to give me a good tank drive feel. You can certainly do this using a pistol grip radio but you would then have to interpret the steering and throttle signals to give you the differential drive action.

Figure 3. MOSFET/relay motor driver.



This is a very simple project. In fact, the controller uses things I've discussed in other articles about creating high power PWM motor drivers using a MOSFET and a relay rather than the more complex, completely solid-state MOSFET H-bridge. This particular board set was designed to be a special effects controller for Halloween spook houses so it needed to be simple and robust. **Figure 1** shows what the board looks like. Note that there are lots of parts on this controller that have nothing to do with the motor controls! **Figure 2** and **Figure 3** show my controller board and motor driver board that use this program. These motor controllers were 7.5 24V maximum.

Creating a fairly high current motor controller is easy with a relay, but you need to pay attention to one really big caveat – DON'T change directions while you are driving the motor! The relay is what handles the motor direction; the MOSFET handles the motor speed. Make sure that you have stopped driving the motor (PWM = 0) before you change direction or you'll get a big fat inductive kick when those contacts break! Sure, it'll probably handle it a couple of times, but

eventually something bad will happen.

In **Listing 1**, my motor speed code clearly shows this "coast rather than suddenly change motor directions" philosophy. This saves wear and tear on everything.

You will only want to implement some of what is on my controller board; this design is a bit much to just drive the motors with an RC radio. The basic design is here for you to pick and choose what you want.

LISTING 1. Motor control code.

```
void MotorA(int chanA)
/*
     Deals with setting PWM for PWM1
*/
{
     signed int speed;
     unsigned int pwm;
     unsigned int chan;

     chan = chanA;

     if (chan > 154)                 //Limit endpoints to be the same for
        chan = 154;                  //both sticks
     else if (chan < 74)
        chan = 74;

     if (chan == lastChanA)
        return;

     speed = 114 - chan;             // positive for forward, negative for reverse

     if (chan > 112 && chan < 116) // create our deadband
     {
        pwm=0;
     }
     else
     {
        if ((lastChanA > 114 && chan < 114) ||
             (lastChanA < 114 && chan > 114))
        {
               setup_ccp1(CCP_OFF); //We coast rather than suddenly
               delay_ms(50);         //change motor directions to save
        }                            //wear and tear on the gear train!
        if (speed < 0)
               output_bit(DIR_1,1);
        else
               output_bit(DIR_1,0);

        pwm = (abs(speed)*3);        //5% to 95%
     }
     setup_ccp1(CCP_PWM);
     set_pwm1_duty(pwm);
     lastChanA = chan;
}
```

Q. I am frustrated from being illiterate on the subject of DC motors and their controllers. This is especially true when it comes to ratings.

All DC motor controller providers give current capacity (10A, 15A, etc.). What about voltage?

Do I need to worry about the motor voltage rating? Do I need to match between a motor and its controller, or will any controller do the job as long as its current rating is high enough to run the motor? Your help is extremely appreciated.

— Amr

A. Those are good questions and ones that I hear fairly often. Matching DC motors to proper motor drivers can be tricky, but there are some pretty simple "rules of thumb" that you can use to simplify the job. I think that you'll find those specification sheets for the motor drivers do indeed specify the maximum voltage that their driver will tolerate, as well as their maximum current. However, just because you will run your motor at or below the maximum voltage and at or below the maximum current doesn't mean that the motor driver chip will do the job that you need. The reason that you aren't in the clear just by being under the specs is: heat — the chip killer. Your L298 driver chip will indeed deliver your 24 volts at two amps for about a minute before it goes into thermal shutdown!

The datasheet for the L298 says that it will handle 50V max at 2A continuous current. Look further into the charts near the bottom of the page, however, and you'll find a chart that graphs "saturation voltage vs. output current." At 2A, that saturation voltage is 2V. Doing a quick power calculation, we find that at 2A this part will be dissipating 2V * 2A = 4W. Ouch! That chip will get hot REALLY fast. The 2A continuous current is the rating as long as the transistor junctions are at ambient temperature (which is 25 degrees C). That junction will NOT be at 25° C for long while the part is dissipating four watts!

To keep that temperature down, you'll need to put on a heatsink. The calculations required to determine the size of that heatsink get very complex and unless you're an engineer designing a printer, you're not going to be interested in them. My advice is to keep your voltage to 1/2 the rating of the driver chip and your continuous current at 1/2 to 1/4 of the rating of the chip; put the biggest heatsink you can fit on it. Run your motors with their typical speed and usage, and test how hot the part gets. If you can hold your

finger on the chip without leaping high into the air and screaming in pain while the motor is running, then everything will be fine. (Of course, you'd have to have the courage to actually touch the chip, and if you're smelling superheated air around the heatsink already, you're probably not going to want to do that.) My second rule of thumb is as long as your current is only 10% of the max for the driver chip, you probably don't need a heatsink.

The most frustrating issue with motor driver specifications is that no two devices will have the same charts and tables. The LMD18200 H-bridge driver chip doesn't have any table that I find particularly useful on its typical datasheet to help determine what is a reasonable level. However, the "Switch On" resistance of the FET driver at max current is specified. In this case, it is a maximum of 0.6 ohms at 3A. Using another power formula ($P = I^2R$), we see that this will dissipate 5.4 watts ($9 * 0.6 = 5.4W$). Again, way too much without a sizeable heatsink! However, the part will simply run quite warm at 750 mA (1/4 of maximum) with a small heatsink. The major saving grace is that we rarely run our robots at full power for a long period of time. If we did, then we'd all be learning those heat transfer formulae to calculate our required heatsink size and configuration.

Hopefully this has helped you out. Here is the your most reasonable rule of thumb for matching a motor to a driver chip: Keep your dissipated power in the chip to less than or equal to half a watt and you'll be okay. This rule of thumb works pretty well for driver chips whose surface area on a side is 2 cm or so. If you are using a motor driver that isn't a chip — for instance, an off-the-shelf motor controller — then the rules are different. You can typically run up to 80% of the rated maximum of the motor controller, for short bursts. We all know that those motor controllers are much more expensive than a simple L298.

Q. I'm having trouble with interfacing my 4x4 keypad to a Zilog microcontroller. I'm using this as input to my robot. I'm using C++ as my programming language. Also, is there any way that I can minimize the use of pins in my microcontroller since I am also using an LCD for displays? It utilized seven pins from my controller. Many thanks!

— Anonymous

A. Finally! A fairly easy question to answer! Yes, you have some options here. Keypads are typically decoded in a matrix where you set an output row and scan the columns to find any "shorts" that indicate a button press. A 4x4 matrix will use up eight I/O lines and a lot of clock cycles. Fortunately, there is an inexpensive option in a common keypad encoder chip that does all this for you. The part to look for is the Fairchild 74C922 4x4 keypad encoder. It is simple to interface and easy to use. The 74C922 will have four outputs whose binary value will be the row/column address of the key pressed. If you want, there is a "Data Available" output that you can check to see if anything is present on the outputs indicating a key press. However, it is probably just as easy to read the data lines; if they aren't zero, then you have a key press. I've found this part at a variety of on-line chip merchants, but the least expensive place to find it reliably is at Jameco Electronics (**www.jameco.com**).

Q. I have built a robot from a wheel chair. I used servos attached to the potentiometer (operated from a Futaba transmitter) so that the servos would move the joystick in the direction it needs to go to operate the two 24 VDC drives. I would like to eliminate this system and put in a speed controller that would operate the two motors affecting movement and steering. With the use of the remote control, can you suggest a system? Thanks.

— Randy

A. This is my month for motor questions! Check out the earlier response in this article where I describe a simple control system that reads the outputs of an RC car receiver to drive two motors in a robot. It is simple and robust.

That said, there are easier ways, of course. My design above uses a microcontroller with the assumption that the robot could be autonomous if desired. If you are only interested in driving the vehicle as a remote control, then you can bypass the microcontroller board entirely and simply connect up various off-the-shelf components that interface directly to your remote control receiver. Some places to find these motor controllers are:

www.trossenrobotics.com
www.banebots.com
www.vantec.com

The Vantech RDFR series controllers, for example, will run two DC motors and even have wires that will connect directly to your RC radio receiver. I recommend an evening's Google session to find the one that fits your budget and needs.

Here's a note from a reader whose question I answered a few months back.

Hi Dennis:

Now that the holidays are over, I can finally get back to my robotic projects.

So, I went to my Dec. issue of *SERVO* and WOW! There it is, my PIC assembly language ADC converter light sensor. Thank you for a great article and the code. I really enjoy the MPLAB system and assembly language is easy to learn. Just wish I had more free time to play. I will let you know how things work out.

Thanks again.

— Bill P.

Thanks for letting me know that my response was what you were looking for! An occasional letter of thanks makes the time put into these articles all worthwhile! PIC assembly is so easy to learn that it's fun. Don't you wish more things in life were like that? Please do let me (us?) know how it all works out. I sure appreciate knowing that our robot builders are succeeding and prospering out there!

Well, that's it for another month of Mr. Roboto! I hope that you are all busily creating more robots to compete with or simply amaze your friends, family, and domesticated animals! Until next month, keep on building! And, if while you are building you have some questions, you know where to find me — roboto@servomagazine.com. **SV**

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

— R. Steven Rainwater

MARCH

7 **CIRC Central Illinois Bot Brawl**
Lakeview Museum, Peoria, IL
This competition has lots of events for both autonomous robots and remote control combat machines including Sumo, mini Sumo, line following, and line maze.
http://circ.mtco.com

12-13 **AMD Jerry Sanders Creative Design Contest**
University of Illinois at Urbana-Champaign, IL
Robots must move balloons around a field to play Tic-Tac-Toe. To make it more challenging, teams must perform specific tasks in order to obtain their balloons.
http://dc.cen.uiuc.edu/

13-14 **METU Robotics Days**
METU Cultural Congress Center, Turkey
A large conference that includes workshops, as well as a wide range of contests like line following, Sumo, mini Sumo, team mini Sumo, slaloming, stair climbing, trash hunting, and robot triathlon.
http://grou.ps/org

20-21 **Manitoba Robot Games**
TecVoc High School, Winnipeg, Manitoba, Canada
This competition includes events such as Sumo, mini Sumo, line following, robot tractor pull, SuperScramble, Frequent Flyer, COGMATION, and Robo-Critters.
www.scmb.mb.ca

20-21 **RobotChallenge**
Vienna, Austria
Events included in this competition are parallel slalom, slalom enhanced, standard Sumo, mini Sumo, and micro Sumo.
www.robotchallenge.at

20-21 **Roboticon**
University of Guelph, Ontario, Canada
Robotic soccer for LEGO robots.
www.collegeroyal.uoguelph.ca

APRIL

8-10 **BotsIQ (previously known as BattleBots IQ)**
Mare Island, Vallejo, CA
In this competition, students build remote controlled combat vehicles.
www.botsiq.org

10-11 **Trinity College Fire Fighting Home Robot Contest**
Trinity College, Hartford, CT
Autonomous robots must navigate a scale model of a house, search for a fire, and extinguish it.
www.trincoll.edu/events/robot

13-15 **DTU RoboCup**
Technical University of Denmark, Copenhagen, Denmark
Varied course including line and wall following.
www.robocup.dtu.dk

15-17 **FIRST Robotics Competition**
Georgia Dome, Atlanta, GA
This is the big championship for FRC teams.
www.usfirst.org

15-17 **National Robotics Challenge**
Marion, OH
This is the NRC Championship.
www.nationalroboticschallenge.org

16 **Carnegie Mellon Mobot Races**
CMU, Pittsburgh, PA

This is the famous annual CMU autonomous mobile robot race.
www.cs.cmu.edu/~mobot

17 **SparkFun Autonomous Vehicle Contest**
Boulder, CO
Autonomous ground and air robots must circumnavigate the SparkFun building.
http://sparkfun.com/commerce/product_info.php?products_id=9016

17 **UC Davis Picnic Day Micromouse Contest**
University of California, Davis Campus, CA
The annual micromouse maze solving contest.
www.ece.ucdavis.edu/umouse

18 **Robot-SM**
Västerås, Sweden
Sumo, mini Sumo, and a robot pentathlon.
www.robotsm.se

22-24 **VEX Robotics World Championship**
Dallas Convention Center, Dallas, TX
This is a really big world gathering of VEX teams for both autonomous and remote controlled competition.
www.vexrobotics.com/competition

23-25 **RoboGames**
San Mateo Fairgrounds, San Mateo, CA
BEAM, Mindstorms, combat, and just about every other form of autonomous and remote controlled robot competition you can imagine.
www.robogames.net

24 **Penn State Abington Fire-Fighting Robot Contest**
Penn State Abington, Abington, PA
Autonomous robots must navigate a scale model of a house, search for a fire, and extinguish it.
www.ecsel.psu.edu/~avanzato/robots/contests

24 **Penn State Abington Mini Grand Challenge**
Penn State Abington, Abington, PA
Autonomous mobile robots navigate an outdoor course.
www.ecsel.psu.edu/~avanzato/robots/contests

24 **The Tech Museum of Innovation's Annual Tech Challenge**
San Jose, CA
This year's competition is called Space Junk.
http://techchallenge.thetech.org

24-25 **Trenton Computer Festival Robotics Contest**
The College of New Jersey Campus Ewing Township, New Jersey
In addition to computer and ham radio events, there are usually several autonomous robot contests. Previous contests included maze navigation, precipice avoidance, and line following.
www.tcf-nj.org

MAY

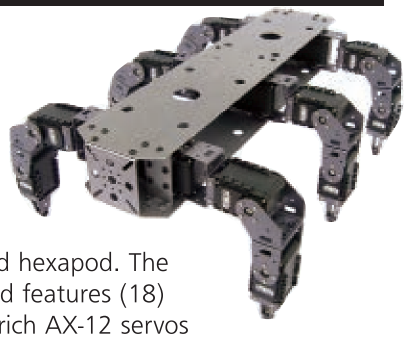
2 **Hawaii Underwater Robot Challenge**
Richardson Pool, Pearl Harbor Naval Station, Honolulu, HI
ROV teams engage in timed, multitasking missions with tethered vehicles.
www.marinetech.org/rov_competition

3-8 **ICRA Robot Challenge**
Anchorage, AK
The challenge includes several events such as Mobile Microbotics, Mobile Manipulation, human-robot interaction, and a virtual manufacturing challenge.
http://icra.wustl.edu

NEW PRODUCTS

CONSUMER ROBOTS

"Smart" Hexapods



CrustCrawler has raised the bar by developing an advanced and capable 3DOF based hexapod. The AX-12-18 Smart Hexapod features (18) powerful and feedback rich AX-12 servos coupled to CrustCrawler's exclusive, all aluminum AX-12 bracket set for leg articulation. The body platforms measure over 15.8 in x 3.5 in (40.13 cm x 8.89 cm) which allows for a large assortment of controlling electronics, sensors, and added hardware.

CrustCrawler has teamed up with Matt Bunting of the Robotics and Neural Systems Laboratory (RSNL) at the University of Arizona to develop advanced walking gaits and six-axis yaw, pitch, and roll control that responds directly to a handheld controller's 3-D orientation. For advanced multithreaded applications such as onboard vision processing, PS3 six-axis control, kinematic calculations, color histograms, face detection, stereopsis, optic flow, and other multithreaded applications, the AX-12-18 Smart Hexapod is mated to a Fit-PC2 computer. The Fit-PC2 is a complete computer based on the powerful Intel Atom Z530 CPU at 1.6 GHz with 160GB SATA harddisk, 1,000 Mb/s Ethernet port, 802.11 WiFi, and six USB 2.0 ports with a low power 12 volt 6W fanless power supply. Complete kits will also feature a wireless Bluetooth USB dongle, six-axis controller, and batteries.

The Fit-PC2 will be available in both Linux and Windows configurations along with free, multithreaded AX-12-18 Smart Hexapod code for both Fit-PC2 platforms. Code will also be developed for complete control via the Internet. A (12) servo AX-12 Smart Hexapod is also in development. The AX12-18 Smart Hexapod will be available in three flexible kit levels:

- Hardware kit – Aluminum body kit only.
- Basic kit – Aluminum body kit and (18) AX-12 servos.
- Full kit – Includes the Basic kit, Fit-PC2 bundle, Bluetooth dongle, and batteries.

For further information, please contact:

CrustCrawler Website: **www.crustcrawler.com**

SENSORS & ACCESSORIES

VRbot — Voice Recognition Module

VRbot Module from Tigal is designed to add versatile voice command functionality to robots; in particular, ROBONOVA-I and ROBOZAK.

The VRbot module can also be used to efficiently implement voice recognition capabilities on virtually any host platform. The module is completely self-contained and interacts with the host through a simple, yet robust serial protocol, enabling voice recognition on relatively low power processors such as ATMEGA, PIC, etc.

Speaker Independent and Dependent Commands

The VRbot module provides users with a host of built-in speaker independent (SI) commands that will allow basic control of the robot movement straight out of the box. In addition, users have the possibility to configure up to 32 user-defined speaker dependent (SD) triggers or commands to control their ROBONOVA and ROBOZAK robots (or other applications developed on an alternate host).

Features

- 26 built-in speaker independent (SI) commands for ready to run basic controls (currently supports US English, German, Italian, and Japanese with more languages available in the near future).
- Supports up to 32 user-defined speaker dependent triggers or commands, as well as voice passwords. SD commands are language independent.
- Easy-to-use and free simple Graphical User Interface (GUI) to program voice commands to your robot or other application.
- Works with ROBONOVA and ROBOZAK MR-C3024 controller boards.
- Module can be used with any host with a UART interface (powered at 3.3V-5V).
- Simple and robust serial protocol to access and program the module through the host board.
- Operates from 9600 bps to 115000 bps
- Sample code provided for the following platforms for a quick start on application development:
  - ROBONOVA and ROBOZAK MR-C3024
  - POPbot Wheeled Robots
  - Arduino USB Boards

– Mikroe EasyPIC5/EasyPIC6
- Protocol specifications provided for implementation on other platforms.

Upon power-up, the supplied VRbotGUI software automatically downloads a roboBASIC bridge program to the controller board which will allow immediate voice control of the robot with the built-in SI command set.

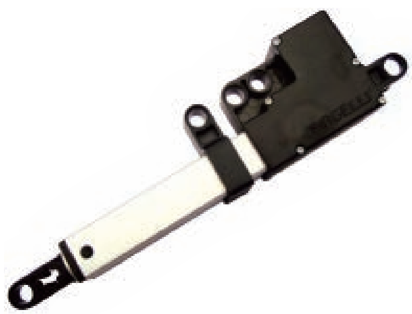
The bridge program also enables programming of the speaker dependent commands into VRbot by using the PC interface. After creating and training all desired commands in the VRbotGUI, users can create a basic template program, open it in the roboBASIC editor, and assign specific behavior to each command. This can finally be downloaded and run the controller, essentially giving the robot the ability to listen and adhere to voice commands.

For further information, please contact:

TIGAL	Tel: **+43 (0)2231/68 347** Tel: **+43 (0)2231/68 402** Email: **info@tigal.com** Website: **www.tigal.com**

SERVOS

LEGO Mindstorms® Compatible Linear Servo



Building on their line of miniature linear servos and actuators for robotics, Firgelli has designed a linear servo specifically for use with LEGO's NXT® Brick. An internal position feedback sensor allows the NXT Brick to control the actuators position to within 0.5 mm. Firgelli has created a custom Linear Actuator Block for NXT-G which allows you to program your creations just as you would a regular LEGO servo motor.

This product — aimed at seasoned NXT pros, students, and enthusiasts — provides a new level of Robotic movement, power, and control to LEGO creations. It operates at nine volts, with speeds up to 12 mm/s or a maximum force of 23N.

The L12-NXT is available in 50 mm and 100 mm stroke lengths. This product is compatible with standard servo motor cables, however, additional adapter cables are available to drive them manually with LEGO Power Functions® and LEGO Technics® components.

A custom software block is provided to allow immediate integration with Mindstorms software. The L12-NXT is also compatible with third party programs such as NXC and ROBOTC. Development of software libraries and drivers for the new linear actuators has already been initiated by several programming experts in the LEGO community. Prices start at $50 each and are available now at Firgelli's website. *(Please note: The LEGO Group does not endorse these products. LEGO, NXT, Mindstorms, Technic, and Power Functions are trademarks of The LEGO Group.)*

For further information, please contact:

Firgelli Technologies	Website: **www.firgelli.com**

PROTOTYPING SOFTWARE

LabVIEW Robotics 2009

National Instruments is offering LabVIEW Robotics 2009, a new version of its graphical system design software that provides a standard development platform for designing robotic and autonomous control systems. NI LabVIEW Robotics 2009 delivers an extensive robotics library with connectivity to standard robotic sensors and actuators, foundational algorithms for intelligent operations and perception, and motion functions for robots and autonomous vehicles. With this new software, users can now implement ideas faster with seamless deployment to real-time embedded and field-programmable gate array (FPGA) hardware, and can maximize the software flexibility through integration with a variety of processing platforms, third-party software tools, and prebuilt robot platforms.

LabVIEW Robotics 2009 is ideal for designing and prototyping applications including the following:

— Autonomous and semi-autonomous ground vehicles
— Robot rescue platforms
— Personal and service robots
— Medical robotic devices
— Academic and research robots
— Agricultural and mining systems

For further information, please contact:

National Instruments	Website: **www.ni.com/robotics**

Show Us What You've Got!

Is your product innovative, less expensive, more functional, or just plain cool? If you have a new product that you would like us to run in our *New Products* section, please email a short description (300-500 words) and a photo of your product to: **newproducts@servomagazine.com**

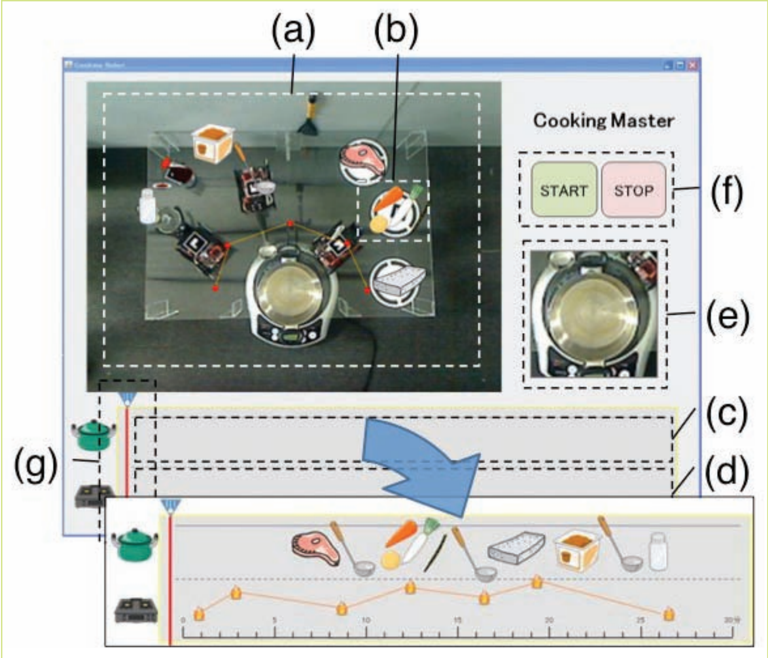
bots IN BRIEF

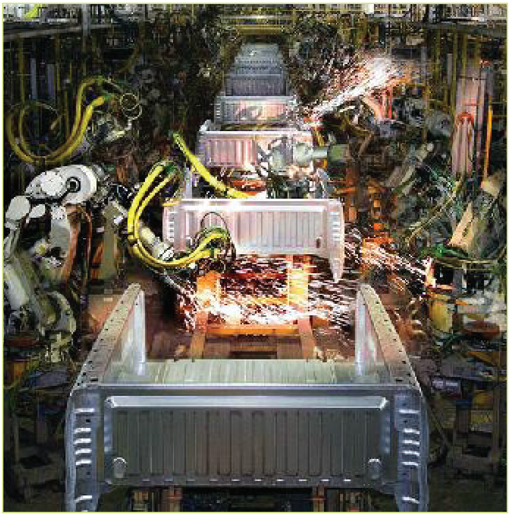
LOOK WHO'S COOKING!

COOKY ROBOTS are the result of one of the research projects undertaken by JST's (Japan Science and Technology Agency) ERATO (Exploratory Research for Advanced Technology) division. The unique thing about these wheeled robots is that it is them doing the cooking work instead of some androids or robot arms that we are used to seeing. The user prepares the ingredients, labels them, and pre-programs cooking time and temperature.

Seems like it would be better if the robots were dealing with 'the preparation of ingredients' part since that is the real work ... not the cooking, itself.

In case you may be interested, the full title of the project is "Cooking with Robots: Designing a Household System Working in Open Environments" and is supposed to be scheduled to appear at the 28th International Conference on Human Factors in Computing Systems (CHI2010) in Atlanta, GA, April 10-15, 2010.





DEATH BY ROBOT?

Thirty-one years ago on January 25th, Robert Williams was killed by a robot arm as it was retrieving parts from a storage facility at a Ford Motor plant. It was the first recorded death of a human by a robot. Williams' family was awarded $10 million in damages due to a lack of safety measures surrounding the robot.

However, Williams' death was described as an "industrial accident," one in which the lack of physical safeguards were at fault. The death was not caused by the robot's will.

THAT'S THE SPIRIT!

There was never any doubt as to whether Spirit would be staying on Mars, but nearly two thousand days of operation instead of ninety? Incredible.

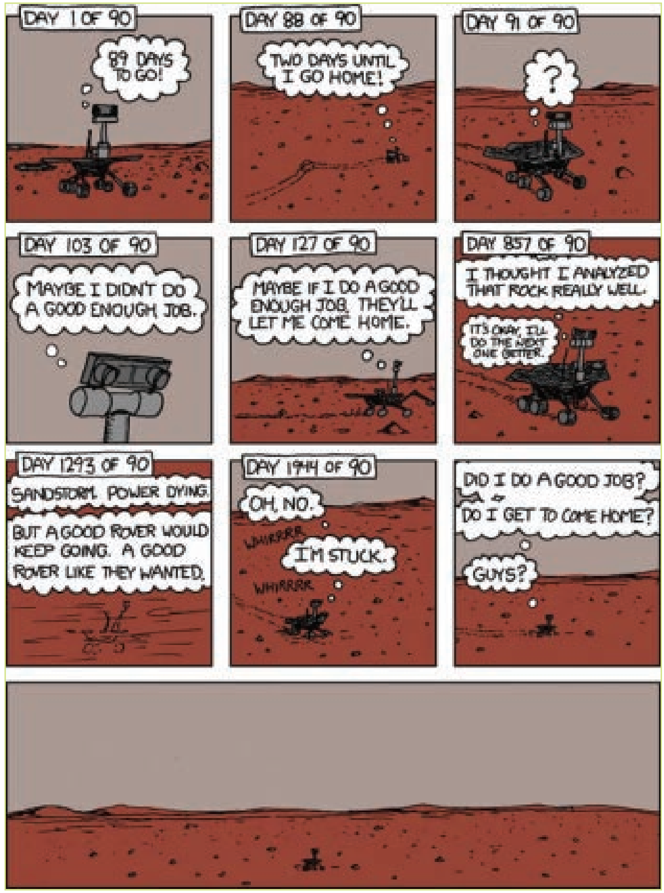
Spirit isn't done yet, but unfortunately, she might be getting close. NASA has officially given up attempts to extricate the rover from deep sand and has declared her a stationary science platform. Over the next several weeks, small movements will be made to try and orient Spirit's solar panels more favorably toward the south, in the hopes that they will be able to generate enough power to keep Spirit alive through the Martian winter.

If Spirit makes it, she could continue to do valuable scientific work for months or even years:

"There's a class of science we can do only with a stationary vehicle that we had put off during the years of driving," said Steve Squyres, a researcher at Cornell University and principal investigator for Spirit and Opportunity. "Degraded mobility does not mean the mission ends abruptly. Instead, it lets us transition to stationary science."

One such experiment Spirit has begun studies tiny wobbles in the rotation of Mars to gain insight about the planet's core. This requires months of radio-tracking the motion of a point on the surface of Mars to calculate long-term motion with an accuracy of a few inches.

"If the final scientific feather in Spirit's cap is determining whether the core of Mars is liquid or solid, that would be wonderful — it's so different from the other knowledge we've gained from Spirit," said Squyres.



Tools on Spirit's robotic arm can study variations in the composition of nearby soil which has been affected by water. Stationary science also includes watching how wind moves soil particles and monitoring the Martian atmosphere.

BYE, BYE BIRDIE

Well, this sucks … in order to pay for Iraq and Afghanistan, the army has been forced to essentially gut the core of its Future Combat System (which has already been gutted a few times) by eliminating the MULE and FireScout unmanned vehicles. One variant of the MULE will remain (a lightly armed version), but the FireScout is toast, to be replaced by an improved version of the Shadow UAV which (for the record) is an unmanned airplane, not a helicopter.

It may make fiscal sense to do this, but it's disappointing. It's sad that one of the most promising aerial platforms I've seen is getting canceled for budget reasons.

Cool tidbits and interesting info herein mainly provided by Evan Ackerman at www.botjunkie.com, but also www.robotsnob.com

I-FAIRY TELLS TALES

The I-Fairy (produced by Kokoro), made an appearance at the recent CES 2010. She can be used as a guide or receptionist as she has voice recognition and goes into her routine when approached by humans. She comes with text-to-speech and motion software, and has a limited 9° freedom in her head and arms (although her wrists, fingers, knees, and toes must be moved manually).

Future owners of the I-Fairy will have the perfect robot to attract people to their booth or to welcome visitors to their event/show. Depending on your needs, I-Fairy could either be completely autonomous (cannot walk) or just rephrase what you are saying through a mic.

Perched atop a pedestal, the Tinkerbot costs approx. $70,000.

ROBOTS UNDER CONSTRUCTION

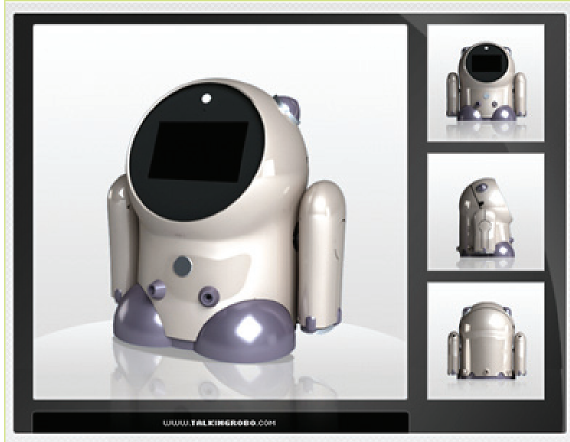
Qatar's Public Works Authority and the Qatar Science Club are cooperating to develop solar powered robots that will help guide traffic around road construction sites. They'll be replacing the people who hold the reversible STOP and SLOW signs when two lanes have been reduced to one by a backhoe or something. This is something a robot would be good at, it's probably going to directly or indirectly put a human out of a job. Hypothetically, giving a robot a job that a robot would be good at allows humans to move on to jobs that humans are better at, which implies something more interesting, (or at least something with more variety).

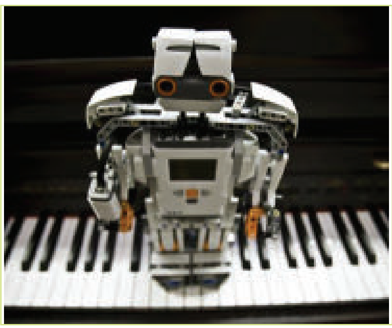
SAY WHAT?

Meet Insultabotz. You can load it up with one or several evil messages up to 15 seconds apiece and let it loose in your workplace or home. When it comes in contact with any object detected by its IR eye — such as people, pets, or water coolers — it will stop, reverse direction, and issue an insult. The six-legged robot is simple to assemble and is no doubt simply annoying.

LOOK WHO'S TALKING!

Now, you can learn a new language with TalkingRobo. With speech recognition, he understands language and will suggest chatting topics. He has face recognition to remember your profile and recognizes images. Show it a picture and he will identify it. It's available in several designs and comes with a dictionary, encyclopedia, and a beam projector. TalkingRobo connects to the Web for updates, making calls, and/or surveillance. Go to **http://talkingrobo.com/** for more info.





NAMES THAT TUNE

A University of Malaga group in Spain, led by Professor Elizabeth Barbancho, devised a LEGO mindstorms NXT that can play Name That Tune when you play chords on the guitar, tickle piano keys, whistle, or hum. The clever bot will then suggest other songs that are similar. A team also came up with a system that uses augmented reality technology to make a virtual keyboard — useful for those without a real one. Other researchers from the group came up with a method of turning ordinary songs into a format that is similar to the one used in "Guitar Hero."

SHOVEL IT!

I-Shovel ('I' for INTELLIGENT) is the world's first and only autonomous robotic snow shovel. With its patent pending technology, I-Shovel keeps your driveway snow free and ready. So, while your neighbors are shoveling their driveways, you can be on your way.

I-Shovel automatically detects the snow accumulation and cleans the driveway automatically. It has a built-in computer to control the shoveling process and various sensors to figure out the perimeter. Unlike snow throwers that are usually used to clean snow after the storm, I-Shovel is designed to wake up and shovel snow whenever it senses a reasonable accumulation of snow. This concept makes it possible to devise a shoveling mechanism with relatively low power consumption and no dangerous moving parts, enabling it to operate unattended. This is a convenience that was only available so far with heated drive ways which are much more expensive to own and operate.

With its three feet wide shovel and 14 inch wheels, I-Shovel is efficient and powerful. It can easily plow through one inch of fresh snow. However, further refinements will be done to the production version of the robot.

The prototype is made with polycarbonate blades, wheels, and handles, and runs on rechargeable batteries with optional solar charging making it eco friendly.

COMBAT ZONE

Featured This Month:

Features

Events

BUILD REPORT:

Hot Stuff Heats Up ComBots Cup!

● by J. Miles

In my opinion, the only thing more fun to watch than two robots fighting it out in a bullet-proof arena is watching two robots with a lot of animation battle it out. So, I challenged myself to build something that would pack as many features as a Swiss Army knife into one fun-to-watch robot that had a goal weight of about 60 pounds. Well, I succeeded! The name of this robot is "Hot Stuff!"

It's a quick, four wheel drive machine with an arm that can grab and then lift other robots completely off the ground. Since holding another robot in the air hardly sounds scary, there is also a flame thrower installed at the end of the arm to torch any robots that become ensnared.

The first thing to do (after buying several cases of Mt. Dew — my official robot building fuel) was to build two drive pods. I found a four foot long scrap of aluminum tube on eBay that was three inches square and had a quarter inch wall thickness — perfect for housing the drive parts! I clicked "Buy it now" and "three to five days" later it was sitting on my doorstep as promised.

I took it out to my robot lab (okay, so it's really just a single car garage) and before you could say "Fighting Robots!" I had it cut into two drive pods with 45 degree angles on each end. As **Figure 1** shows, I took each drive pod and clamped them into my milling machine to cut rectangular openings for the drive wheels to stick out of.

Figure 2 shows a drive pod

FIGURE 1

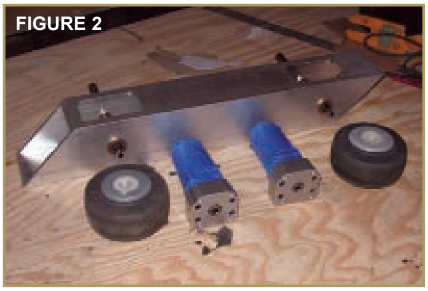


FIGURE 2

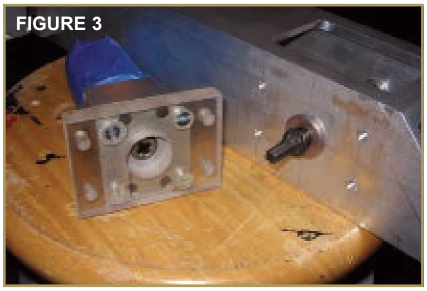


FIGURE 3

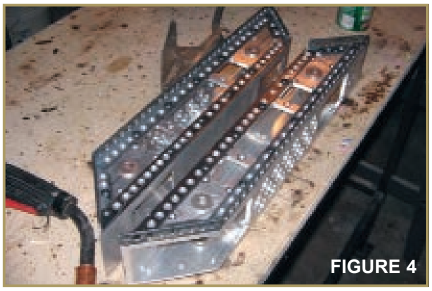


FIGURE 4

with the holes for the wheels all cut out, as well as the other parts that make up the drive unit. The robot will be four wheel drive with a motor and gear box in each corner (that was extracted from an 18 volt Dewalt cordless drill). You can see the guts from two of the drills (in **Figure 2**) with blue painters tape over the motors to keep stray bits of metal from getting sucked inside and onto the magnets.

For tires, I used four wheels that were originally meant to be caster wheels on some type of cart. They were very grippy, relatively light weight, and came with a bore already set up with a key way to transfer the drive power from the half inch drive shaft to the tire.

Since the bright yellow cases that the drills came in didn't offer an easy way to mount them in a robot, I got rid of them. Instead, I used an aluminum mount from **Teamdelta.com** combined with a clear mount that I made from polycarbonate to attach the motors to the aluminum drive pods. In **Figure 3**, you can see the motor with the mounts, the bronze bushings I used for drive bearings, and the half inch drive shaft with a funky (and obviously male!) spline that fits into the female spline-shaped output that the Dewalt drill motors have.

I was trying to cram a lot of robot into 60 pounds, and I knew early on it was going to be tight on weight. **Figure 4** shows the two drive pods after I basically turned them into Swiss cheese! It takes a lot of holes in quarter inch thick aluminum to lose any weight. (For example, it takes 135 holes that are 5/8" diameter to lose one pound!) Also in **Figure 4** you can see the first parts of the steel frame that connect the two drive pods together, bolted to the aluminum drive pods (and like the drive pods, they are also full of holes!).

You can see the robot starting to take shape in **Figure 5**; when you look at it from the side it resembles a parallelogram. The majority of the robots that fight in robot combat rely on some type of heavy blade that spins at 2.4 billion miles per hour to build up a bunch of kinetic energy, and then they hit your robot with said blade usually ripping large chunks of it off. To combat these spinners, I hoped to use the angled front and back ends of the robot to deflect any hits from these monsters.

Figure 6 really shows the robot coming together. I used 1/8 inch thick by 3/4 inch wide cold rolled steel strapping Mig welded together to build most of the frame. I drilled lots of holes in some of the frame members and left some of them solid, depending on how much force I anticipated each piece would see in combat. I personally don't like looking at metal boxes fighting each other, so I tried really hard to put lots of interesting angles into Hot Stuff.

I struggled for a while to find a small enough fuel tank to power the flame thrower since most of the tanks I found where meant for portable torches used in plumbing. These tanks were both too large and too heavy. I settled on a butane fuel tank and valve from a camping stove to power the flame thrower on Hot Stuff. I cut the rubber tubing from the camping burner and plumbed it to a 24 volt solenoid valve that I could turn on and off by remote control. I used another length of tubing to run from the solenoid valve to the front of the lifting arm so the flame would come out there. To ignite the gas, I used an electronic sparking unit meant for a cooking BBQ that turns on whenever the gas valve opens (see **Figure 7**).


FIGURE 5

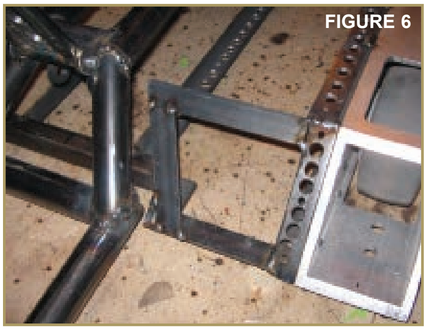


FIGURE 6


FIGURE 7


FIGURE 8

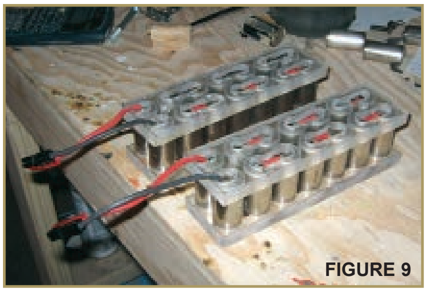


FIGURE 9

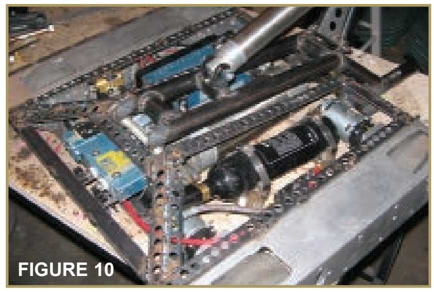


FIGURE 10

The competition I built Hot Stuff for would have matches that lasted three minutes long or until one of the robots got knocked out (whichever happened first). Balancing battery power can be tricky because if your battery is too small, then you risk running out of juice before the fight is over. If your battery is too big, then you are just packing around extra weight you could be using somewhere else. **Figure 8** shows me making a battery pack housing that will space the cells apart to keep them cooler.

Heat is an enemy to anything electrical; in battery packs, extra heat can cause the packs to run less efficient and run out of juice faster. To combat heat, I built cell spacers out of polycarbonate to keep each Sub C cell an 1/8 inch from one another. I used Sanyo 2400 Sub C sized Nickel Cadium battery cells to power Hot Stuff because they are lightweight, can source a lot of power, and they can be charged right in the robot. **Figure 9** shows the two battery packs finished and ready to go in the robot.

Instead of rambling about how I mounted all the internal parts by bending this little piece of metal here or welded that tab there, I'll just show you the photo with most of the internals mounted (**Figure 10**). I'm using a paintball tank that holds nine ounces of CO_2 to power the lifting arm. I run the CO_2 through a regulator that takes the 900 or so PSI (pounds per square inch) of pressure that's in the bottle and lowers it to 150 PSI.

There are two air cylinders in the robot: a larger one to run the lifting part of the arm and a smaller one to run the grabbing part. Each air cylinder is run by a valve hooked to an R/C switch so I can control it with a radio transmitter designed for R/C airplanes.

In **Figure 11**, you can see the robot stripped down to just the frame. I hung it from my robot lab rafters (Okay, okay, my garage rafters) with a little piece of wire and hit it with some red spray paint. This part of the robot ended up only weighing 13 pounds — not to bad!

This was definitely going to be a very animated robot because *every* channel on the radio receiver was full (see **Figure 12**)! It was actually a really big pain to get all the plugs to plug in at the same time. (I guess the manufacturer figured they would never all be used at once.)

I started putting the robot back together after the paint dried. I used some black spray paint on the drive pods (I really liked how it all came out). In **Figure 13**, you can see I started to wire up the speed controller. This is one of the most important parts of a fighting

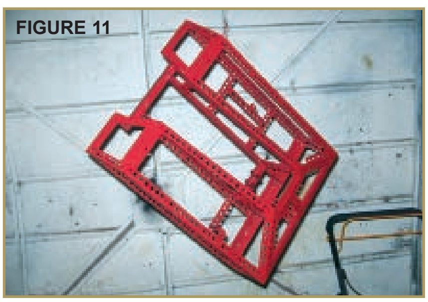


FIGURE 11

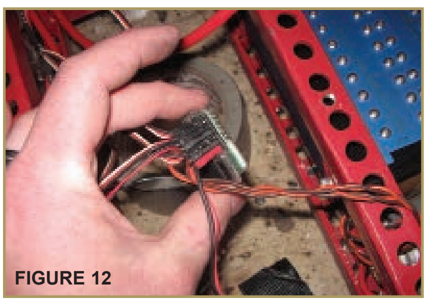


FIGURE 12

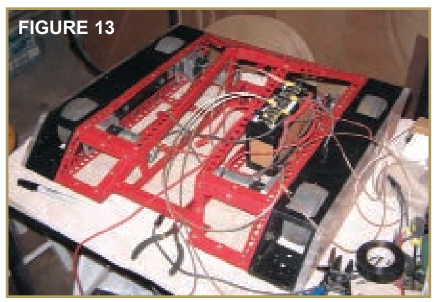


FIGURE 13

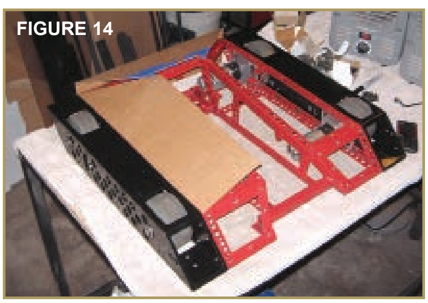


FIGURE 14


FIGURE 15

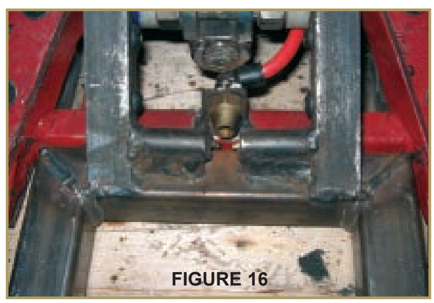


FIGURE 16

robot because this is what you wire your batteries, motors, and radio receiver to so that the robot goes where you tell it with your remote control.

To armor the robot, I chose polycarbonate which is a clear plastic that is literally bullet proof in some thicknesses. I used 1/8 inch thick pieces for the top and back armor, and 1/4 inch thick for the front (I figured this would see more action). I used cardboard to make a template of the armor and then traced that onto the plastic to cut it out with my band saw (see **Figure 14**).

To build the grabbing and lifting arm, I used mild steel and aluminum (since my local Home Depot was out of un-obtanium). The lifting arm has two pitchfork tongs that drag on the ground to scoop under other robots. The grabbing arm has a spike on the end made from three little triangles of steel to grip other robots so they can be picked up off the ground and held in the air. On each front corner of the robot, there is also a spike made from titanium; these can be used as ramming weapons but their main purpose is as outriggers to keep Hot Stuff from face planting when it picks up another 60 pound robot. You can see the finished robot minus the top armor panels in **Figure 15**.

Figure 16 shows a close-up of the nozzle the flame comes out of and the gas line that feeds it. The nozzle is actually one that goes on the tip of a blower for an air compressor. If you look closely, you can see the two small copper wires that come out of the round tubes welded to the lifting forks. These are run to the BBQ sparker and provide a nice blue spark a few times a second to light the gas.

FIGURE 17

Figure 17 (taken by Michael Mauldin of Team Toad) shows Hot Stuff competing at the Combots Cup in the 60 pound weight class against Frosty the SnowBot. Hot Stuff had some close fights but was able to get several robots up in the air and shoot lots of fire! After all was said and done, Hot Stuff went undefeated and took first place! SV

MANUFACTURING: Pattern Routing of Plastic Parts

● by Pete Smith

I produce limited production runs of some of my combat robot designs and these often require fairly complex flat plastic parts (**Figure 1**). The easy way to do these parts is to have them watercut by a company like Team Whyachi (**www.teamwhyachi.com**), but if you need parts quickly and your time is cheap, then with a little careful design you can route the same parts using just a router and a pattern.

The first thing to do is make an aluminum pattern of the part. A friend machined mine on his CNC mill, but it can also be done by hand or be watercut. If you have an existing part, then it could be also used as a pattern. The smallest, easily available pattern bit is 1/2" in diameter, so the smallest internal radius on the parts must be 1/4" or greater. All other required

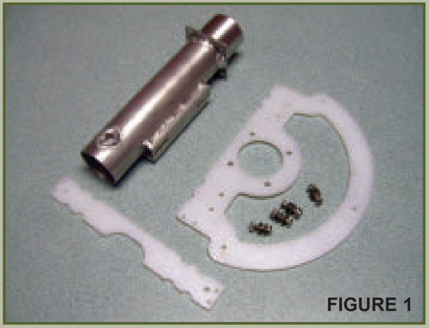


FIGURE 1

FIGURE 2

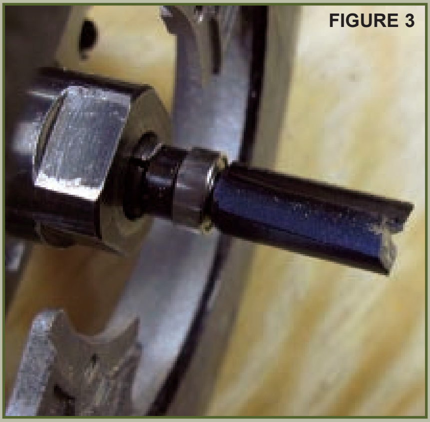


FIGURE 3


FIGURE 4

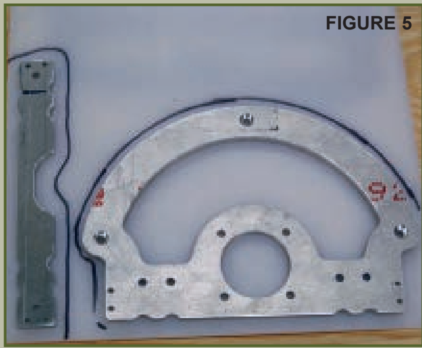


FIGURE 5


FIGURE 6

mounting holes should be included in the pattern so that they can be added to the part after it is routed. Extra countersunk holes were added so that the pattern could be securely fixed in place. The patterns for two of my parts can be seen in **Figure 2**.

The routing bit (**Figure 3**) has a 1/2" diameter ball bearing just above the cutting blade. This bearing will contact and follow the pattern and as the blade is also 1/2" in diameter, the part will be a copy of the pattern.

WARNING*: Router cutters are very sharp and must only be handled using thick leather gloves or you will get severe cuts (learn from my mistakes!). Safety glasses must also be worn.*

The router bit has a 1" long cutting depth, so it is necessary to build a jig that will allow the blade to cut the thin 0.2" UHMW (Ultra High Molecular Weight) without always having to cut 0.8" of supporting material each time. I made the jig (**Figure 4**) by gluing four 24" squares of 1/2" plywood together. The squares were cut to size for free by the local DIY store and secured together using Liquid Nails® adhesive. To make sure they were pressed together firmly while the glue set, I parked one of the wheels of my car in the center of the square!

I then screwed the patterns onto the jig and (with the router depth set so the bearing would run on the pattern) then routed out a clear channel around the pattern and removed all the material within the openings. The router I used was a cheap 1-3/4 HP model from Harbor Freight. It does the job, but the collet used to secure the cutting bit was poor quality and took some packing with thin paper to get it to grip the cutter adequately. I would buy a better quality machine next time.

The UHMW comes in quite large sheets so in order to make the parts easier to handle, I outlined some oversize blanks (**Figure 5**) and cut them out with a jigsaw. These were then sandwiched between the pattern and the jig, and then secured in place with woodscrews (**Figure 6**). I set the router bit height again so that the bearing would contact the edge of the pattern; the perimeter of the parts were then routed (**Figure 7**). The router makes quick work of this, but be sure to maintain a firm grip on it and keep it level throughout the cuts.

The interior openings are done in the same way, but first you drill through the center of the area with the end of the router bit. This completes the router work and the remaining small holes can be added using appropriately-sized drill bits and the pattern as a guide.

With a little practice and care, high quality parts can be produced very quickly. I made three complete sets of parts is about an hour (**Figure 8**).

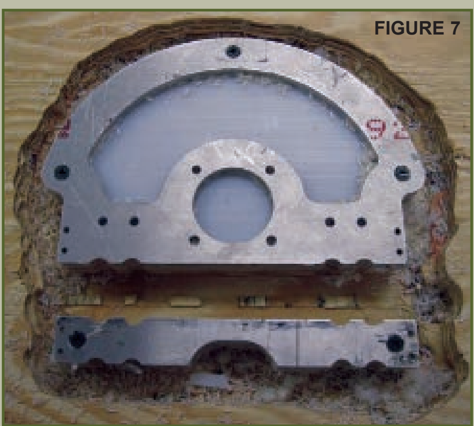


FIGURE 7

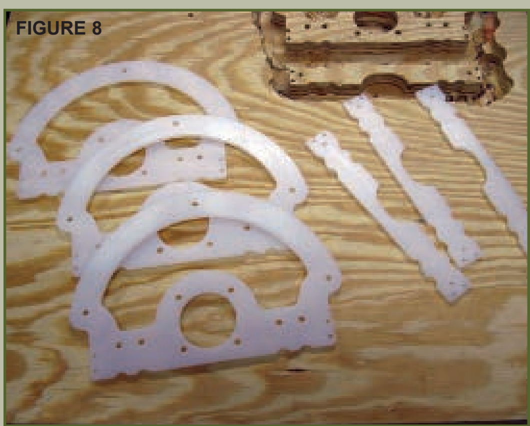


FIGURE 8

My next beetleweight design requires two sides which are near identical. Using the above method, all that I need to do is create one pattern and then I can produce both sides and as many spares as I want from that pattern, reducing both the time and cost of building and repairing my bot. SV

PARTS IS PARTS: Lipo Puffometer

● by Nick Martin

Lithium polymer batteries are increasingly popular in robots. In combat bots, they are often abused to the point of failure with expensive and dangerous results. Before a pack fails or burns, it often 'puffs' to larger than its normal size. This is a sign that the pack is either losing capacity or is likely to burn under a normal current drain. Puffing is not always obvious and to check my Lipo packs, I invented the Puffometer.

How to Make It

The Puffometer is very easy and cheap to make. You will need a drill, jigsaw, and some scrap plastic. The material can be almost any sort of plastic sheet as long as it is fairly rigid; in **Photo 1**, the Puffometer is made from an off-cut of Garolite. Measure the height and width of your pack, then cut a square about one inch larger all around. Draw a rectangle with the dimensions of your pack in the center of the plastic and then use a drill and jigsaw to cut out the rectangle. I found that most of my packs had larger dimensions at the wiring end; you should check dimensions at the middle of the pack for a good fit in the Puffometer. Check the battery fits through the cut-out, clean up the edges, and you're done. If you use more than one type of pack, simply use a longer piece of plastic with multiple holes; just remember to label each hole!

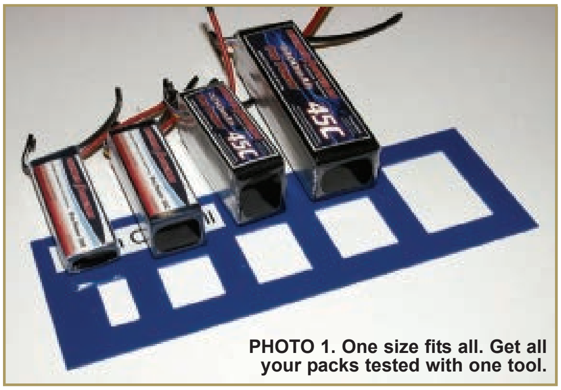


PHOTO 1. One size fits all. Get all your packs tested with one tool.

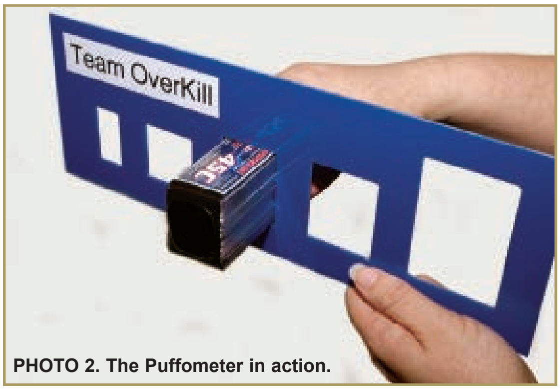


PHOTO 2. The Puffometer in action.

How to Use It

To use the Puffometer, wait until your Lipo pack is at room temperature. Slide the pack at least halfway through the correct hole — it's that simple! If the pack does not fit as easily as usual, it is starting to puff up and needs to be treated carefully. If the pack does not fit at all, you need to discard the pack immediately. SV

EVENTS

Completed and Upcoming Events

Completed Events: Dec 8, 2009 to Jan 9, 2010

The ComBots Cup IV was presented by ComBots on December 19th and 20th in San Mateo, CA.

Gulf Coast Robot Sports-4 was presented by Gulf Coast Robot Sports in Bradenton, FL on January 2nd.

Upcoming Events for March - April

Central Illinois Bot Brawl 2010 will be presented by Central Illinois Robotics Club in Peoria, IL on March 6th. Go to **http://circ.mtco.com** for more information.

2010 Robot Rumble will be presented by Carolina Combat

Robots in Durham, NC on March 20th. Go to **www.carolina combat.com** for more information.

RoboGames will be presented by ComBots April 23rd through 25th in San Mateo, CA. Go to **www.robogames.net** for more information. SV

EVENT REPORT: The ComBots Cup 2009

● by Ray Billings

Over the weekend of December 19-20, 2009, the ComBots Cup event was held in San Mateo, CA where combat robots ranging from one pound up to 220 pounds competed for honor and glory. The action was fast and furious in every weight category, but the main focus of the event was on the 220 pound heavyweight class, as contestants faced off to compete for the ComBots Cup trophy.

The ComBots Cup event is the pinnacle of today's heavyweight combat robotic action which also has the possibility of a sizable cash prize. The ComBots Cup was founded by Dave Calkins in 2005. As a robotic enthusiast, Dave appreciates all forms of robotic endeavors. As a promoter, he understands this simple truth: Audiences prefer watching the larger combat machines. In an effort to actively encourage involvement by larger combat machines, he wanted to pick a single weight class to promote with a large cash prize. Cash prizes at the ComBots Cup have been as high as $10,000! He reasoned that this would spur new machines in the category, and promote a higher level of competitiveness overall. With dwindling numbers in the 340 pound super heavyweight category — but relatively strong numbers for the heavyweights — the decision was made to provide some incentive for that category and the ComBots Cup was born.

The ComBots Cup is similar to the Stanley Cup in hockey. The ComBots Cup is a piece of history, and a living, growing record of championship. Unlike trophies from other combat robotics events, a new ComBots Cup trophy is not made each year. With each event, the previous winner must surrender the Cup to the new champion. The winner's name and robot image are then engraved into the base of the trophy. By winning the Cup, you become part of that history. At any one time, there is only one ComBots Cup champ, and being the holder is an honor indeed!

Currently, the event is run and managed by Simone Davalos, who is also the referee. This is how she describes the event: "The vision of the ComBots Cup is to reinvigorate the big bot heavyweight class. We

PHOTO 1. Sewer Snake with its tough lifting wedge takes on the overhead spinning bar of Brutality.

PHOTO 2. Avalanche with its 360° rotating plow faces off against Original Sin.

PHOTO 3. All three previous ComBots Cup champions competed at this event. Shown here left to right are: Ray Billings with Last Rites, Paul Ventimiglia with Brutality, and Matt Maxham with Sewer Snake.

PHOTO 4. One of the crowd favorites of the show was the area with big bots on display. Here some audience members are taking a closer look at the internal parts of Vlad the Impaler 2.

PHOTO 5. One of the more brutal matches of the event was the tough wedge of Sewer Snake against the big spinning bar of Last Rites.

PHOTO 6. Combat events always involve frantic repairs between matches. Here, Matt Maxham does double duty, repairing Sewer Snake while being interviewed on camera.

know it's an expensive sport, so we offer a chunk of prize money to help offset the cost and make the event more fun for competitors. We have been very encouraged since the start of the ComBots Cup to see so many competitors coming out of the woodwork for some events, and we're always excited to see more. A lot of what we do with the trading cards and builder meet and greets is inspire more kids and adults to build something themselves. They can always find out how to get involved at **Combots.net** and **Robogames.net** after they come home from an event."

As evidenced by the overall quality of competition in this weight class, plus the new heavyweight machines that have entered over the last few years, it's obvious that the ComBots Cup has indeed had an impact on the heavyweight class.

The very first Combots Cup was held in November 2005 at the Fort Mason Center in San Francisco, CA. This event included most of the active heavyweights of the time plus a few new machines, and also marked the return to combat of some veterans that hadn't competed in many years (such as the infamous Biohazard). The action was furious and intense, but in the end Matt Maxham's creation Sewer Snake took home the Cup in the inaugural event, narrowly beating out Donald Hutson's machine Karkas 2 in the finals. Matt's renowned driving skill proved to be the deciding factor in winning the event.

The second ComBots Cup was held May 2007, in conjunction with the Maker's Faire at the San Mateo, CA County Expo Center. Paul Ventimiglia's overhead bar spinner, Brutality, took home the Cup that year beating out Gary Gin with his tough wedge, Original Sin, in the finals. Brutality had taken some major damage in this event, breaking his spinning bar during a match against the big offset spinning bar of Last Rites. Paul showed his ability to adapt by shortening his weapon bar to balance out the damage, and by outdriving his opponents throughout the tournament to secure the victory.

In May 2008, the event was held again in conjunction with the Maker's Faire. It was run as a round robin tournament which resulted in some intense matches as each entrant had to face off with every other opponent in the class. I am proud to say that my creation, the offset spinning bar machine Last Rites, ended up taking home the 2008 Cup. I would love to say that my great driving skill was the deciding factor, but truthfully it was more a matter of luck. That and the pure determination it took to keep my machine running in the face of such high quality opponents.

Last year's ComBots Cup was held in December 2009, at the San Mateo County Expo Center. The field at this year's event was a little smaller than in the past, but the quality of the machines was top notch including all three previous ComBots Cup winners. The tournament was once again run as a round robin event, and given the overall quality of the machines competing, it was no surprise that the action was as intense as ever. At stake was a prize purse of $5,000 and, of course, the bragging rights of being the ComBots Cup champion. In the end, the extremely tough wedge Original Sin — expertly driven by Gary Gin — took home the 2009 Cup. Gary went undefeated, toppling all previous ComBots Cup champions to earn his place in history.

If you would like to be part of the action — either in the audience

PHOTO 7. Here we see Michael "Fuzzy" Mauldin signing trading cards for some enthusiastic fans.

PHOTO 8. With its tough design and Gary Gin's excellent driving, Original Sin is always a top machine at any competition. Here, we see Gary with Original Sin and the ComBots Cup trophy.

or as a participant — the next all weight class combat event will be at RoboGames, scheduled for April 23-25 at the San Mateo County Expo Center. For further details, go to **www.robogames.net**. SV

Photos were taken and provided by Team Toad: Michael "Fuzzy" Mauldin, Kelsey Ross, and Sharon O'Mara. Title photo is courtesy of Sam Coniglio.

ZOIDBERG: Thinking Outside the Box

● by James Baker

Project Zoidberg was conceived when a motor delivery for one of our other robots arrived at my teammate's house, containing a surprising 'extra' component. The sender had accidentally packed a pair of Team Delta18V Dewalt 'power drive' gear motors. These powerful, drill-based units looked too nice to send back, so a deal was arranged. Not wanting to build another boring armored box, the seeds of new 30 lb robot were sewn.

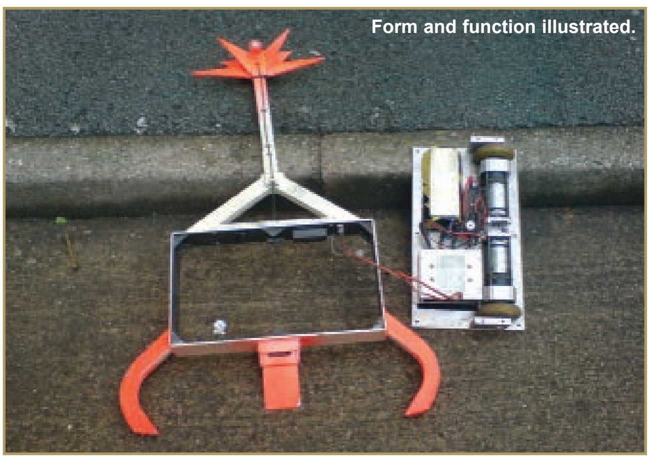
Function

When designing the Zoidberg robot, it was immediately apparent that this was potentially the most capable drive system used in any of our featherweight robots. Looking at the technical specifications for the motors, it was calculated that a single pack of high capacity NiCad 'C' cells would suit the current requirements. A 40 Mhz radio system is used to control the robot (as we have a few spare), and the obligatory activation light and removable link were added to conform to UK rules. The electrical power system uses black and red high temperature silicon wire throughout.

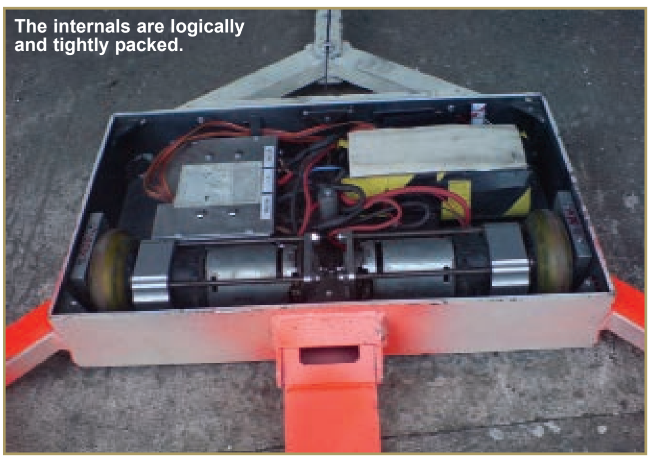
A wooden base was made to test the power drives — directly driving 80 mm scooter wheels using custom-made hubs. These hubs were machined from high-grade aircraft aluminium, and bonded to the wheels with a special resin. From these first tests, it was decided that even though many people run these gear motors on 24V we would run the robot at 18V, as the performance was still very good compared to the majority of machines in the UK at the moment. The lower voltage also offered improved reliability and longevity while we got used to using the robot.

The controller chosen was the Robot Power Sidewinder. This is an over-specification solution, and allows us the option of increasing to 24V in the future without compromising reliability. We also have Sidewinders in several other robots, so spares are available if we have failures at events. The initial tests with the drive system showed a surprising amount of grip was available from the scooter wheels, so although a four wheel drive solution was investigated, it was not considered necessary (especially when the incredible agility of the two wheeled chassis impressed us during these first tests).

Once testing with the wooden base was complete — impressing us with its speed, agility, and having pushed our 220 lb axebot across the workshop without slowing down — the final specifications were frozen and the components were transferred onto a base made of 6 mm aircraft aluminum. The construction of this robot would be done in two distinct stages: first the interior, then the exterior. The shape and concept of the exterior would



Form and function illustrated.



The internals are logically and tightly packed.

be defined by the weight and size of the internals.

The priority at this stage was to package the components as tightly as possible. This tight packaging would allow the motive part of the robot to be small and strong, and give greater freedom to the external concept. The optimum packaging resulted in a layout that was low in height, short in length, but quite wide in order to have both motors inline at the front of the chassis.

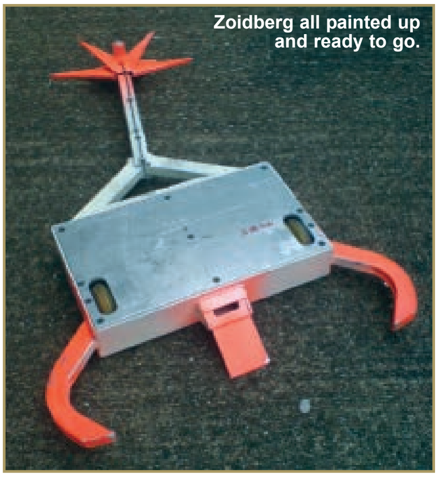
Once the internal section was finished, the temptation would normally have been to use up the remaining weight allowance by adding an active weapon or thick armor and maybe a wedge front, making it as competitive as possible, but we've done that before. Although this approach would have been effective, we felt that something more visually interesting was needed for this particular robot.

Form

There was a significant amount of weight still available once the inside of the robot was finished. Many concepts were discussed, but as we wanted something interesting and elaborate, for survivability purposes it was decided that this robot would not have active weapons and would be fabricated from easily repaired mild steel.

The first task was to protect the internals. For this we used 4 mm mild steel, and created the dreaded box shape. Looking to nature for inspiration, the basic shape of a scorpion suited the strengths of the robot — claws at the front to grab opponents, and a tail at the rear, to swing as an impact weapon.

The claw shapes were developed by trial and error, tack welding different versions to the steel inner box, with the final versions seeming to be good against most of the shapes available to test on. These claws effectively corral opponents, trapping them and allowing Zoidberg to use its power



Zoidberg all painted up and ready to go.

and grip to take the opponent wherever it wants. Between the claws, there is a small hinged wedge designed to get underneath the opponent irrespective of which way up Zoidberg is at the time, and lift the opponents off its wheels — further increasing Zoidberg's drive advantage.

At the other end of the robot, a hinged bar with a heavy array of spikes proved to be a useful tail. Different tail lengths of a similar spiked design were tested, and although the shorter versions were more practical, the longer ones looked so naturally dangerous that the final two versions are on the longer side of ideal. Having tested the tail on our own combat robots, it does more damage than you might expect from a passive clubbing weapon.

The fact that the tail is hinged to the robot adds an organic element to the way the machine moves, creating swooping arcs and a momentum during direction changes that induce sideways drifting. The tail also acts as a stabilizer when returning to 'straight and level' driving, which means we do not need a gyro to help control the robot.

Once the robot was complete, it needed painting. Looking at the robots we already have, many are dark colors such as blue and black. Though cool looking, we have found they are difficult to see in the arena

Zoidberg's two part concept illustrated my teammate Peter Willson-Hughes.

sometimes. For Zoidberg, a base coat of bright white was highlighted at the extremities by luminous orange. This bright but simple paint scheme lends itself to drawing attention to the robot in the arena, but as the robot is primarily designed to crash into the opponent at speed, the color is simple enough to repaint rapidly.

Zoidberg has been tested at a number of events to-date, and has proven itself to be a potent machine already. However, certain upgrades have been necessary. The aircraft grade aluminium wheel hubs have been replaced with stainless steel versions after identical failures on both of the old ones. A new tail had to be fabricated as one of the original two "disappeared" at a science festival event run by a university. Finally, with reliability and performance proven, the upgrade to 24V was decided upon, using 3700 mAh NiMH 'sub-C' cells.

As a final point of interest, the name Zoidberg was conceived early on when the initial plan was to fabricate a lobster inspired shape. It was not aggressive enough for our young driver, Daniel, so the concept was altered to that of a scorpion (even though Daniel wanted to keep the name). SV

From Brazil to San Francisco

By Marco Meggiolaro

FIGURE 1. RioBotz at RoboGames 2006.

Our Incredible RoboGames Experience

Our combat robot team RioBotz can be divided into two timelines: B.R. and A.R. – Before RoboGames and After RoboGames. We have been competing with combat robots since 2003, starting with a lousy (though beloved) middleweight named Lacraia, that had an “amazing” 1 mm thick aluminum armor (so as not to go over the weight limit). This all started when a few of my college students asked me to help them build a combot. I had built robots before, but never to be sent into combat or to have a strict weight limit. This was new to both me and my students. After a lot of learning, broken parts, and reading all available books on combots, we achieved our first significant result the following year, winning the Brazilian nationals with the middleweight spinner Ciclone.

Back then, Brazilian competitions were very different from today. Most robots weren't powerful enough to inflict any major damage on adversaries. The rivalry was very intense, not only among teams but also between the various states around Brazil. The cheering was sometimes too passionate, resembling a street fight crowd. A few teams insisted on covering their robots in the pits (as if, during the competition, there would be time for competitors to modify their robots to take advantage of any obvious weak spots). The competition was fun, but there was not too much sharing of information.

Things changed after Paulo and Thacia, the organizers of Brazil's largest combat event (the Brazilian equivalent of Dave and Simone), attended RoboGames 2005. They had a wonderful experience — completely different from the Brazilian events back then. Not only were the combots more powerful and destructive, but the builders were much more mature and friendly. Paulo and Thacia gathered a lot of information about building combots and pit etiquette, and shared it with the Brazilian community on their return.

They also shared several videos from RoboGames 2005 which inspired many teams to attend the event the following year (especially our own RioBotz). But the combat videos were scary! Most Brazilian robots were not even close to that level! Our relatively unstable middleweight spinner Ciclone would be flipped over for sure against any well-built wedge, and its 1/4" thick, low-grade aluminum structure would be cut like butter by powerful spinners.

This was when there was a huge upgrade in the RioBotz robots. RioBotz had been winning most Brazilian events with a middleweight robot that was not invertible and whose weapon energy was the same as the one from our current featherweight. We were lucky – we never had any major damage inflicted on us. In fact, you only learn your robot's limits when something breaks. Our spinning bar broke once in Brazil, which led us to improve the weapon system. Unfortunately, our armor was still lousy by RoboGames standards. To change that, we started designing Touro in late 2005. It would be an invertible middleweight combot with a powerful spinning drum. We also designed and built our first beetleweight combot, Mini Touro, to compete at RoboGames 2006.

Our sponsors didn't have the money to send the team to RoboGames 2006, so all the students who attended had to pay for their own ticket and hotel room. This is pretty much how it still is today, since it's not cheap to compete overseas – but

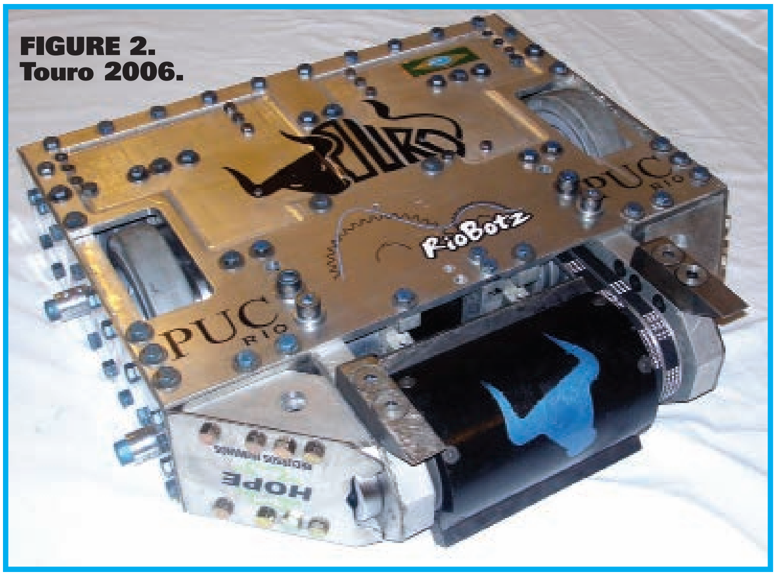


FIGURE 2. Touro 2006.

FIGURE 3. All Brazilians competing in 2006.

Figure 4. Front armor.

Figure 6. Touro vs. Sub Zero.
(Photo by Brian Benson)

it is well worth it! **Figure 1** shows the RioBotz team at RoboGames 2006 — only seven of the 15 students on the team could afford to travel with us. Due to final exams just before the trip, we had to finish the robot at the San Francisco hotel. (I do not recommend that! But at least it was fun.) There was a "mechanical systems" hotel room and an "electrical assembly" room. All grinding was done at the hotel parking lot. **Figure 2** shows Touro on our hotel bed, having finally been completed. Other teams from Brazil attended RoboGames 2006 as well, and are shown in **Figure 3** during a welcome party.

RoboGames 2006 was a blast. It was the first time ever we watched live heavy and super heavyweights fight. At that time there were just middleweights in Brazil. We were very well received by all the teams in the pits. We were pleased to see everybody showing their robot's details, even when we would be facing them next. We gathered a lot of information on motors, mechanical parts, speed control tweaks, and so on. It was a crash course in robotics for sure!

We also learned from our mistakes. For instance, in the 2006 Touro vs. Ice Cube match, one of our two battery packs failed due to a broken solder that occured right after the first impact. Touro finished the round very sluggish, costing us the match. Since then, we've only used battery packs with flexible copper braid instead of rigid tabs. We highly recommend this.

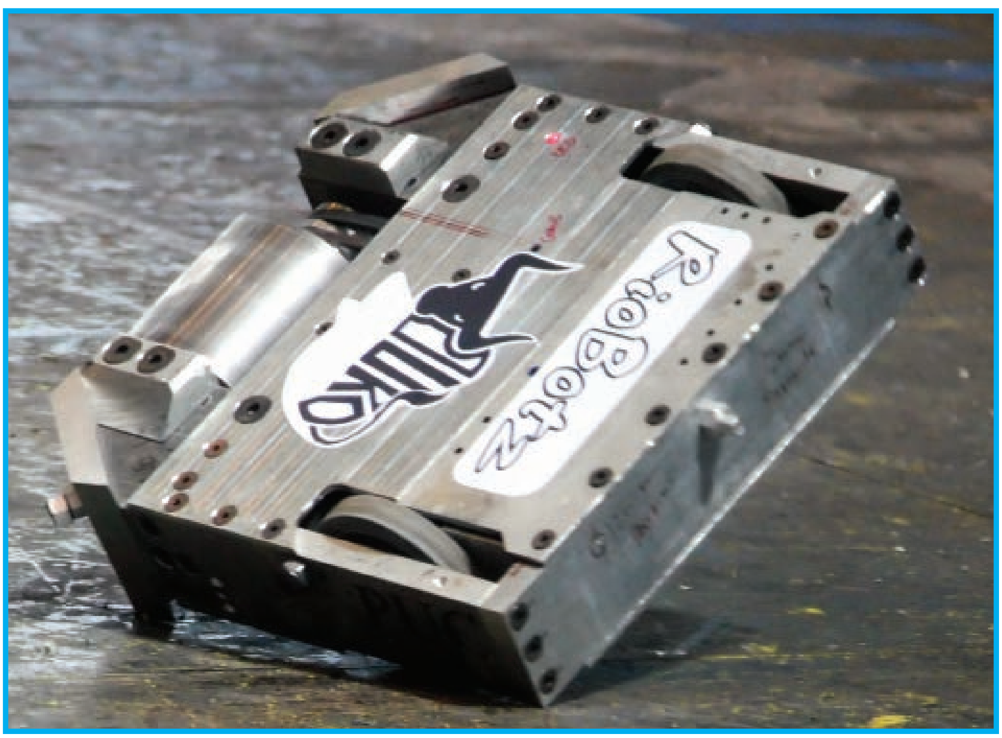
We also learned a lot from our next match against Mortician. Even though we won the match by KO, it took us eight hours to fix the bot afterwards. We still used hex head bolts back then, and many were sheared during the fight. Mortician hit our thin titanium front armor plate on a surgical strike exactly where it was not supported by the aluminum structure (**Figure 4**); this almost destroyed our weapon system. Since then, we've used thicker armor plates there, and only flat or round head Allen screws.

The whole team was excited to watch the other competitions. There was so much to see and learn that the three days were just not enough. This is when the A.R. era began.

The whole experience was so enlightening that I started writing the RioBotz tutorial as soon as I was back in Brazil. There are several great build reports, forums, books, and tutorials showing how to build combots, however, there was nothing written in Portuguese (translated excerpts from my tutorial have been printed here in *SERVO's* Combat Zone). My idea was to not only write about building combots (including all I had learned at RoboGames 2006), but also to describe the great experience I had there. I wanted to tell Brazilians that robot competitions are not only about robots and winning, but also about knowing people and sharing knowledge. There is a whole chapter about RoboGames and pit etiquette which has had a huge impact on the Brazilian community.

In the first six months of putting my tutorial on-line, there were 10,000 downloads! I believe every Brazilian combot builder became aware of RoboGames and its friendly environment. The following year's Brazilian competitions were noticeably different — hardly anyone was hiding their

Figure 5. Touro Light 2007.
(Photo by Brian Benson)



robots under covers. People were proudly showing their bots in detail, and many teams were helping each other fix their robots. This was uncommon before 2006. I also believe the tutorial helped to increase the number of Brazilian teams who compete, not only due to the technical tips, but also due to the description of our wonderful experiences at RoboGames. Finally, the fact that we won both a gold and bronze medal in 2006 also had a great impact. It showed other teams that it was possible for a Brazilian robot to win, even in such a tough event.

Figure 7. Touro vs. Mortician 2009.
(Photo by Brian Benson)

RoboGames also had an influence on the number of categories in Brazilian hosted events. Since our attendance at RoboGames, we added hobbyweight combat in 2007, featherweight combat in 2008, and then followed by 3 kg Sumo and hockey in 2009.

Since 2006, we've attended RoboGames every year. For RoboGames 2007, we increased the speed of Touro's drum. We needed that extra energy against the increasingly well armored bots. The width of the S7 drum teeth was increased from 3/4" to 1" to take all the impact energy. All hex screws were changed to flat or round head, so they wouldn't be torn off. We also built our first lightweight — Touro Light (**Figure 5**) — using the same idea as Touro. **Figure 6** shows one of our most exciting matches — Touro against Sub Zero. In only our second year at RoboGames, we went undefeated in both middle and light classes to win two gold medals. It was a wonderful experience!

Even though our robots didn't break, we've learned a lot from the fights. We had to change Touro Light's drive motors after every match — using only two 775 motors was not enough for a lightweight. After facing Pipe Wench, we realized that it would be good for Touro if its drum was reversible, to effectively work even when flipped over.

We also realized that our robot's ground clearance was too low, since it got stuck quite often. In Brazil, our arena does not have to take the punishment of heavy or super-heavy bots, so its floor is very smooth (with seams no greater than 1 mm). If you have a match right after The Judge or Megabyte, several seams may have opened up. We've realized that the bot has to adapt to the arena, and not the other way around.

For RoboGames 2008, we decreased the thickness of Touro's top and bottom plates to gain a little ground clearance. This didn't solve the problem, but it did help a little bit. Touro's drum became reversible after changing the weapon solenoid to a pair of Victor speed controllers. Touro Light's drive motor was changed to an 18V DeWalt. This led to the birth of our first featherweight: Touro Feather.

Touro was doing well until it got defeated twice by the vertical spinner Professor Chaos. One of our main problems was our 1" wide tooth which did not withstand a weapon-to-weapon hit while Touro was flipped over. Touro Light didn't do well either – its planetary gearboxes didn't take the higher power from the 18V DeWalt, shearing the pins from the last stage in two different matches. Touro Feather had a very powerful brushless motor spinning the drum, however, the drum's v-belts were too rigid which prevented the brushless motors from spinning up sometimes. The low starting torque of brushless motors can be a problem if the BESC (brushless electronic speed controller) is not well programmed.

In 2009, I released the English version of the RioBotz Tutorial. I had learned so much at the RoboGames 2007 and 2008 events – both from other builders and from broken parts – that the tutorial doubled in size, reaching 380 pages. You learn a lot from these types of events. It's like drinking water from a fire hose sometimes, so be ready to take several pictures and notes.

For RoboGames 2009, we increased Touro's S7 drum tooth width to 1.5", using a new drum with a higher moment of inertia. Both Touro and Touro Light's drums featured a single S7 tooth, balanced by a tungsten counterweight flush with the drum. Touro Light's drivetrain was upgraded to a pair of DeWalt power drives. Touro

FIGURE 8. Touro Light vs. Come to Mamma.
(Photo by Brian Benson)

Figure 10. Touro Maximus vs. Sewer Snake.
(Photo by Brian Benson)

Feather had its weapon motor switched to a brushless with a lower speed constant to increase its starting torque. Touro's higher energy drum and single tooth did quite a bit of damage on its opponents. However, Touro was sent to the loser's bracket by Mortician (**Figure 7**), and in another fight got stuck in the arena due to its low clearance.

Touro Light delivered some nice hits with its single toothed drum (**Figure 8**), but its drive motors overheated so much they caught fire. We talked later with the guys from Texas Heat and found that we should have used a current limiter between the Victors and the DeWalts since we were over-volting them quite a bit.

Touro Feather was doing okay until the double KO against Death by Translation (**Figure 9**) ripped all the drive motors from their gearboxes and made the weapon's brushless motor pop open. The robot was entirely rebuilt that night for the following day's match, but it was still crippled and did not do well.

Touro Maximus — our first heavyweight robot — was built specifically to compete at RoboGames 2009. It is a wider version of Touro, with almost the same length and height. Unfortunately, it didn't do well against Original Sin and Sewer Snake (**Figure 10**) and their skillful drivers.

We also debuted our 3 kg Sumo bots which won two gold and two silver medals! Their development was an excellent lesson in fuzzy logic for the team. We competed in fairy, ant, beetle, and auto-beetle categories.

For the upcoming RoboGames 2010, we've decided to entirely redesign the Touro family — from middle to hobbyweights. The main issue with the previous versions was their very low clearance. They also used too many screws which was a pit repair nightmare.

We're currently experimenting with a new design. Touro and Touro Light's entire chassis has just been waterjet cut from a 3" thick aluminum plate. In this way, each bot's chassis only consists of two parts, making it a "bi-body" (a two-part unibody). This decreases the number of screws, while making the robot stronger. Both top and bottom plates are still separate parts. We've increased the size of all the screws, in order to use fewer of them and make the pit stops faster. The new chassis will leave a higher ground clearance to avoid getting stuck in the arena. The waterjet design allowed us to make the bots with a curved shape, eliminating several stress raisers. The same idea is applied to Touro Feather and to our hobby Touro Jr. The new profile should make it possible — depending on the drum energy — to self-right the bot by just using the gyro effects. It's already working with Touro Feather. I'm not sure how effective they will be, but at least they'll look cool!

This April will mark our fifth year at RoboGames. We'll be competing in all combat classes except super-heavy, entering both autonomous and remote controlled 3 kg Sumo. We'll also have the international debut of our Hockey Pro Team. We've come a long way in four short years, and it's been a tremendous growth experience — not to mention all the friends we've made.

We hope to see you *all* at RoboGames 2010! **SV**

Figure 9. Touro Feather vs. Death by Translation.
(Photo by Brian Benson)

Robotics Showcase

Explore the New ZeroG Technology to Go RoboWireless

By Fred Eady

Gathering data is a meaningless process unless you establish a method of transporting and obtaining your data on demand. There are many ways to cultivate data these days. You can use the Internet, IEEE 802.15.4, point-to-point radio, Ethernet, RS-232, or USB. The data gathering method you choose depends on the amount of and type of data you are collecting. For instance, if you are monitoring a bunch of robotic devices that are equipped with sensors, you would need to go with 802.15.4, 802.11b or Ethernet. A single command and control robotic device could get away with point-to-point radio access. In any case, if you have a data movement requirement that involves using the Internet, you'll also need to call upon the services of Ethernet.

Everybody talks about setting up Ethernet networks. However, few actually detail their experiences. So, this month, we'll take a step-by-step look at setting up a wireless Ethernet network using the Microchip ZeroG technology. By the time you read the last sentence, you'll know how you can use the ZG2100M in your robotic applications.

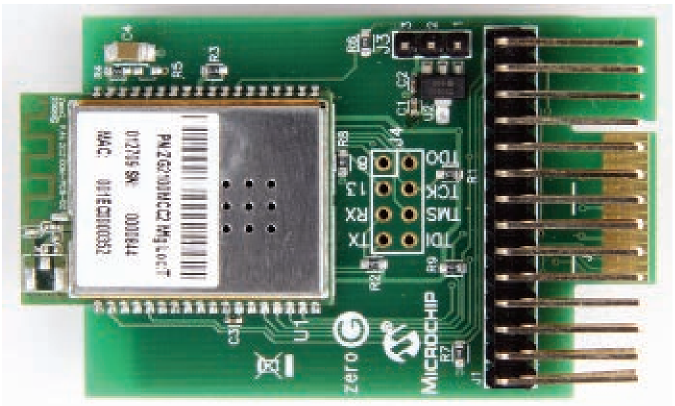
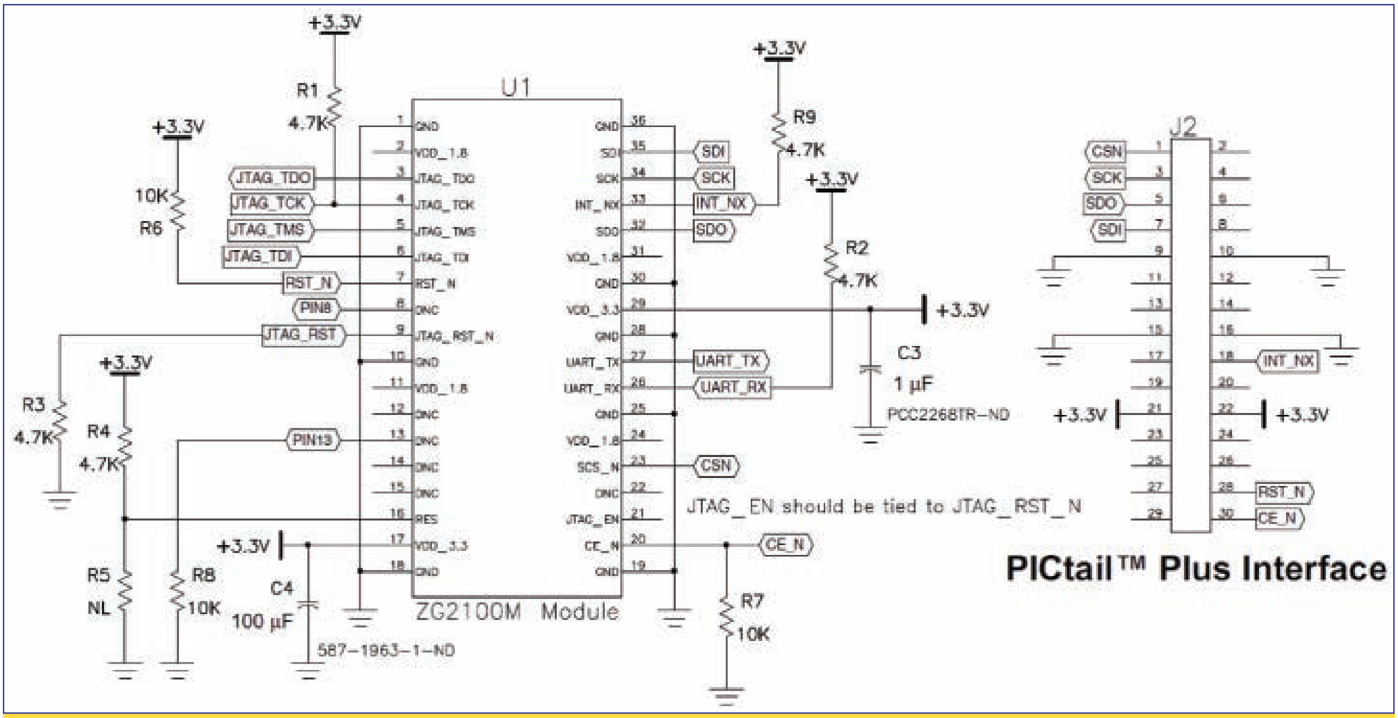


PHOTO 1. This is a shot of the ZG2100M PICtail which is configured to be attached to the Explorer 16 development platform. Note the ZG2100M's printed circuit board antenna.

ZeroG

Recently, Microchip acquired ZeroG. Along with the company came a new wireless Ethernet device that is well-suited for robotic applications. ZeroG's ZG2100 is a single-chip 802.11b Wi-Fi transceiver that supports data rates up to 2 Mbps. The words mobile, sensor, and robot all go together. The ZG2100 adds yet another term to that list: low power.

The ZG2100 chip is not sold in its native state. Instead, the ZG2100 chip is incorporated into the ZG2100M module which you see in **Photo 1**. The ZG2100M is integrated into a PICtail configuration. PICtails are designed to be mounted on Microchip Explorer 16 development boards. The advantages of using a PICtail and development board include a quick proof of concept and the ability to attach multiple PICtail devices to the same Explorer 16. For instance, I used a development board loaded with a USB-equipped PIC24FJ256GB110, an ENC624J600 100 Mbps Ethernet PICtail, and a USB PICtail to develop a USB-to-Ethernet application for the Microchip 2009 MASTERS



SCHEMATIC 1. As you can see, the ZG2100M interface is simple. A master SPI portal, three command output pins, and a data interrupt input at the PICtail Plus Interface controls the movement of data and the state of the ZG2100M.

conference. The Explorer 16 development board/ZG2100M PICtail hardware we'll be talking about here is smiling for the camera in **Photo 2**.

The Hardware

The Explorer 16 posing in **Photo 2** is equipped with a PIC24FJ128GA010. A ZG2100M PICtail can also be seen standing tall in the Explorer 16 development board's interface socket. Once we get this ZG2100M hardware configuration loaded up with the ZeroG firmware driver, we can migrate the design to a PIC platform of our own design.

The ZG2100M consists of a ZG2100 802.11b transceiver, all associated RF components, a crystal oscillator, the necessary glue components, and a printed circuit board antenna. Just about any eight- or 16-bit microcontroller with a native or emulated SPI portal can utilize the ZG2100M's radio facilities. With that, the UART pins you see in **Schematic 1** are intended for use as a debug interface only.

When mounted on an Explorer 16, the ZG2100M is powered by a 3.3 volt supply. The ZG2100M can assume a number of modes or power states. Thus, it is important to understand how the ZG2100M's CE_N pin's logic state affects its power consumption and internal operation. Obviously, when both the VDD_3.3 and CE_N pins are not powered, the ZG2100M module is considered to be void of any power. Applying 3.3 volts to both the VDD_3.3 and CE_N pins forces the ZG2100M into hibernation mode. When it's hibernating, all of its internal circuits are turned off which means that an external device must be used to coerce the ZG2100M to enter and exit hibernation. Sleep mode is entered when the ZG2100M is powered and the CE_N pin is held logically low. The ZG2100M's reference clock and internal bias circuitry are enabled when it's

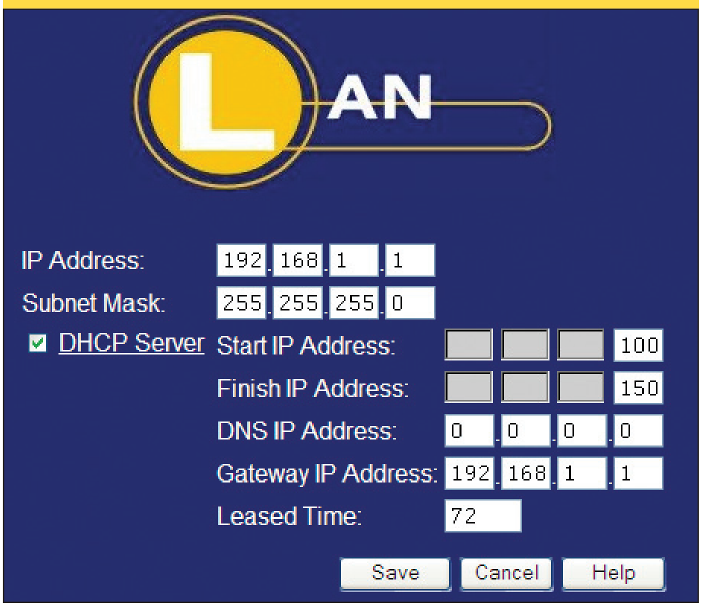
PHOTO 2. The Explorer 16 development board is loaded with a PIC24FJ128GA010 and ZG2100M PICtail. The Microchip engineers have provided ZeroG code within the Microchip TCP/IP stack to support this configuration.

PHOTO 3. This Access Point has seen a couple of MASTERs classes and has logged many an hour on the bench.

sleeping. Again, an external stimulus from a controlling device must be present to bring out the sun, moon, and stars for the ZG2100M module.

Since even the smallest microcontroller can execute an instruction in less than a microsecond, it is important to know how much time it takes for the ZG2100M to awaken from a power-up and move between states. The ZG2100M needs 300 mS to stabilize after power-up. When transitioning to and from hibernation, it requires the controlling microcontroller to allow 50 mS for the transition to complete. The latency for enabling the ZG2100M's receive circuits is 0.2mS. Receive to transmit and transmit to receive switching latency is 0.01 mS. The ZG2100M wakes up and enters a standby state in 0.2 mS. The standby state is a transitional mode only. Thus, when the ZG2100M wakes up, it takes 0.2 mS to enter standby mode and another 0.2 mS to enable the receive circuitry.

SCREENSHOT 1. The ZG2100M configuration requires that our Access Point provide a DHCP server function. This is the screen that turns on the Access Point's DHCP server.



Power consumed by the ZG2100M in sleep mode is 250 µA. The sleep power mode is managed by the ZG2100 and allows for keeping in touch with the access point to which it is associated using the least amount of power. During hibernation, the ZG2100M draws a miniscule 0.1 µA of current. These very low power consumption figures make the ZG2100M perfect for battery-powered applications. Outside of the realm of the Explorer 16 development board, the ZG2100M can operate on voltages between 2.7 and 3.6 volts.

The ZG2100M's SPI portal is configured as a slave. From the viewpoint of a SPI slave, the SPI master is responsible for providing the clocking that is necessary to move data between the slave and master devices. The SPI clock speed is limited to 25 MHz. The master SPI device also has the responsibility of activating the target slave device if other slave devices are under its control. To alert the master of data availability, the ZG2100M includes an interrupt output line that is driven logically low when the slave ZG2100M contains data it needs to forward to the master device. Once the master empties the ZG2100M's data buffer, the interrupt line is driven logically high by the ZG2100M. Another look at **Schematic 1** reveals that the SPI master portal on the Explorer 16's PICtail Plus interface includes a ZG2100M slave select line (CSN), a ZG2100M reset output (RST_N), and a CE_N state/mode control output.

Another very important feature of the ZG2100M hardware is the inclusion of a valid IEEE MAC address which is factory coded into every ZG2100M module. According to the ZeroG datasheet, each ZeroG module MAC address is in the range of 001EC0xxxxxx.

Okay. You're trained on the ZG2100M. Let's move on and start setting up a ZeroG wireless network.

Configuring the AP

The AP (Access Point) you see in **Photo 3** is a babe in the woods as it knows nothing about the ZeroG network we want to build. So, we'll begin our Access Point's education by teaching it how to serve IP addresses via its internal DHCP server module.

The AP defaults to an IP address of 192.168.0.1. That piece of IP information allows us to use Internet Explorer as our educational portal. Simply entering http://192.168.0.1 in the Internet Explorer address frame gives us access to the AP's internal configuration engine. There is another network on the bench running in the 192.168.0.1 domain. So, I modified our ZG2100M network's AP to use the 192.168.1.0 IP segment. **Screenshot 1** is representative of the configuration page that we used to enable the DHCP server. Note the new LAN IP address and that our pool of IP addresses now begins at 192.168.1.100 and ends at

192.168.1.150.

Now that our AP's DHCP server is activated, we must provide the ZG2100M a target to shoot at. That target is the wireless network SSID (Service Set Identifier). The SSID that the ZG2100M is in search of is coded in the Microchip TCP/IP stack's TCPIPConfig.h file:

```
// Default SSID or wireless network name
// to connect to
#define MY_DEFAULT_SSID_NAME    "MicrochipDemoAP"
```

After modifying the AP's default SSID value to match the firmware configuration, we need to check the Broadcast SSID box in **Screenshot 2** which tells the AP to include the SSID in the informational packets it transmits. The ZG2100M picks up on the broadcasted SSID and attempts to associate with the AP that transmitted the SSID it has been coded to mate with, which in this case is *MicrochipDemoAP*.

If you plan on using WEP, you'll need to configure the AP accordingly. Take a look at **Screenshot 4**. To keep things as simple as possible, I've allowed the AP to automatically select between Open System and Shared Key authentication. The 128-bit WEP key was generated using a Passphrase of "SERVO."

Right now, that's all we need to configure as far as the AP is concerned. Networking is a lot like algebra. What is done on one side of the equation must be equivalent to what is done on the other side. So, let's move to and configure the other side of our network equation.

Configuring the ZG2100M Firmware

As you've probably already figured out from our AP configuration exercise, the ZG2100M is also capable of implementing common security methods such as WEP. Let's walk through the steps that are necessary to activate WEP security for our ZG2100M module. The code we will be modifying is contained within the Microchip TCP/IP stack's TCPIPConfig.h file. The TCPIPConfig.h file contains logical switches and definitions that will determine how the TCP/IP stack is configured and built in the compile phase.

The first thing we need to do is tell the TCP/IP stack which security method we want to use. This is done by altering the MY_DEFAULT_ENCRYPTION_TYPE definition:

```
#define MY_DEFAULT_ENCRYPTION_TYPE
kKeyTypeWep
```

The TCP/IP stack allows the use of both the 64-bit and 128-bit WEP key structure. I prefer to use the longer 128-bit structure. So, let's set that up:

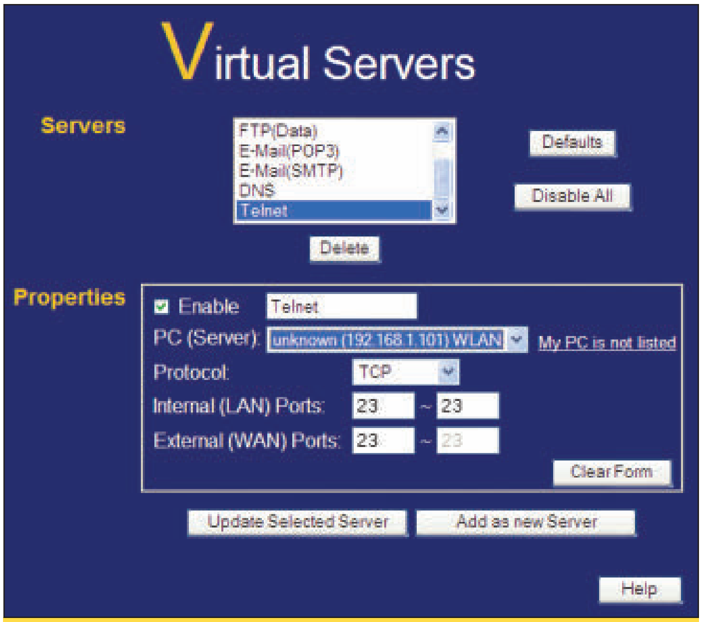
```
#define MY_DEFAULT_WEP_KEY_LEN
                          kZGWEPKeyLenLong
#define MY_DEFAULT_WEP_KEYS
                          MY_DEFAULT_WEP_KEYS_LONG
```

SCREENSHOT 2. In addition to advertising the Access Point's SSID, this configuration screen also allows us to set the channel of operation and the 802.11 modes to use. The ZG2100M is based on 802.11b, and since we're only interested in communicating with the ZG2100M, we have selected to use 802.11b mode only.

Naturally, the next step involves entering at least one WEP key that matches the one logged into our AP. All we have to do to enter a WEP key is replace one of the default key strings:

```
#define MY_DEFAULT_WEP_KEYS_LONG        {
{{0x91,0x3A,0x78,0x49,0x90,0x94,0xC0,0x0B,0x37,0x
0F,0xB3,0x0F,0x1D}},\
```

SCREENSHOT 3. I used the phrase "SERVO" to generate this 128-bit WEP key sequence.



SCREENSHOT 4. Most every Access Point has a facility to allow Internet users to access servers that sit in the Access Point's local network.

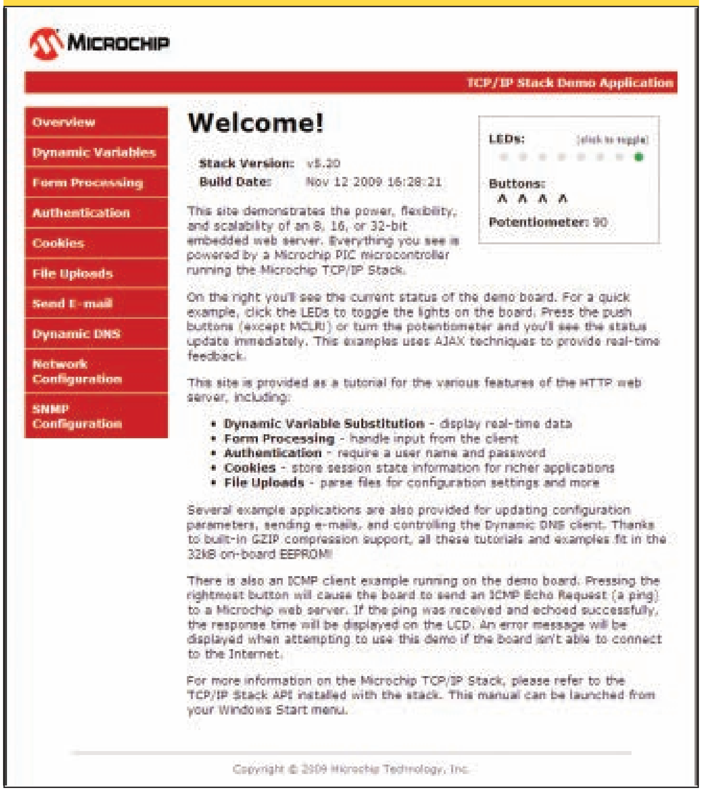
```
{{0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x
19,0x1A,0x1B,0x1C}},\

{{0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x
29,0x2A,0x2B,0x2C}},\

{{0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x
39,0x3A,0x3B,0x3C}},\

}
#define END_OF_MY_DEFAULT_WEP_KEYS_LONG
```

SCREENSHOT 5. The Microchip folks that engineered the TCP/IP stack have taken much of the head scratching out of embedded networking. This set of Web pages is a fantastic tutorial on sending mail, uploading files, and coding interactive Web pages.



Note that the 128-bit WEP key I entered matches the AP's WEP key entry in **Screenshot 3**. There are four key index strings that we can alter. I chose to replace the first WEP key string to match the first WEP key string in our AP. The TCP/IP driver needs to know that. So, we identify the selected key index in this manner:

```
/* Valid Key Index: 0, 1, 2, 3  */
#define MY_DEFAULT_WEP_KEY_INDEX      (0u)
```

The final step in setting up ZG2100M wireless security is defining the WEP authorization type:

```
/* Valid WEP auth:   kZGAuthAlgOpen    */
/*              kZGAuthAlgShared */
#define MY_DEFAULT_WEP_AUTH         kZGAuthAlgOpen
```

Take a quick look at **Screenshot 3** again and note that I selected *Auto* in the AP's Authentication configuration slot. Selecting *Open* would have also worked here as we can specify the authentication type in both our TCP/IP firmware and the AP configuration.

Now our ZG2100M data flow is relatively secure. If your ZG2100M-equipped mechanical animals roam on your residential range, you'll most likely want them to be able to get on the Internet and your local network through your current (and hopefully secured) home network setup. That's where being able to configure the ZG2100M for WEP and other security methods become important. Now you know that you can monitor and control your ZG2100M-based devices while keeping your home network and device data secure.

Here's where the cool stuff comes into play. Take a look at **Screenshot 4**. The PC(Server) is really our ZG2100M which has been given the IP address of 192.168.1.101 by the AP's DHCP server. The *unknown* prefix is telling us that a NetBIOS name was not associated with the device (our ZG2100M) that accepted the lease to the DHCP-assigned IP address. Not a problem. In this application environment, we can operate without a NetBIOS name.

What we've done in **Screenshot 4** is open up an Internet worm hole in the AP to allow our ZG2100M to act as an Internet-enabled Telnet server. A remote Internet host can access the ZG2100M Telnet server application which is part of the free TCP/IP stack and is running on the PIC24FJ128GA010 by opening a Telnet session using the AP's IP address coupled with the port addresses shown in **Screenshot 4**. The same can be done if you wanted a

Fred Eady can be reached via email at fred@edtp.com

remote Internet host to access the ZG2100M Web server application at port 80. In fact, you can assign your own "garage-brewed application" a port number and allow Internet access to it using TCP and/or UDP.

Speaking of Web Applications

The ZG2100M segment of the TCP/IP stack contains a precompiled Web application that I think you would be interested in. If you are looking to build interactive Web pages, upload files, or send mail using your ZG2100M, the tutorial pages in the ZG2100M demo area will get you started.

I loaded the ZG2100M Web page demo onto our Explorer 16 Development Board/ZG2100M platform and dialed up the resultant HTTP server at 192.168.1.101. The index.htm page you see in **Screenshot 5** welcomed me. The upper right corner of **Screenshot 5** is an interactive pane that displays the status of the Explorer 16's pushbuttons and LED array. You can also interact with the development board's LEDs and pushbuttons from this area using your mouse.

Go RoboWireless

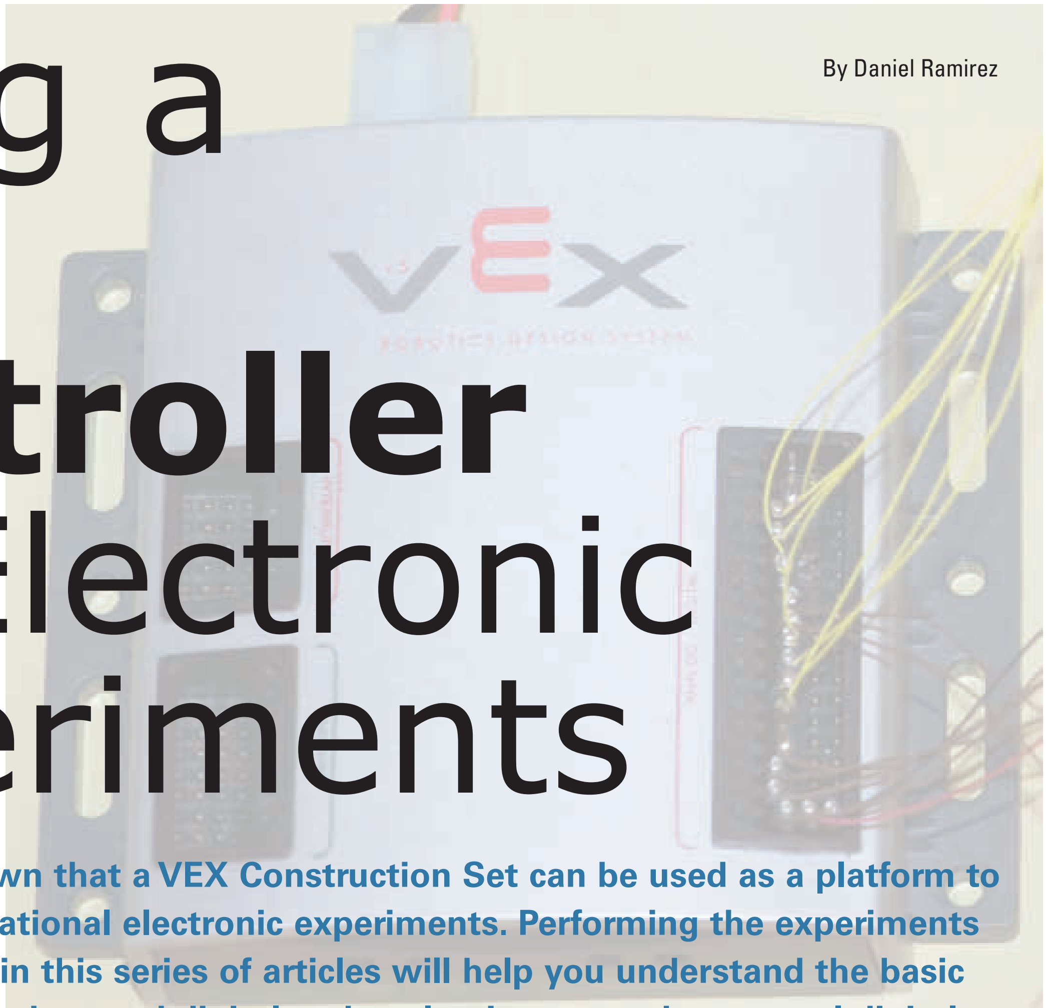
Think about it. Now your grazing piece of iron can send you an email detailing what he or she has encountered or whining about having low battery voltage. You might want to use a ZG2100M to create a roving interactive Web server. The ZG2100M hardware is easy to interface and most of the firmware and magic RF stuff has already been taken care of. All you need to use your noggin for is to engineer the overall application from the ZeroG hardware and firmware building blocks at your disposal.

The point is that you now have access to what I call Star Trek technology. How many "wired" anythings can you remember seeing on Star Trek? Utilizing the ZeroG wireless technology we've just discussed and a simple PIC, you can become a modern-day Scotty. **SV**

ZeroG ZG2100M
Microchip TCP/IP Stack
PIC24FJ128GA010
Explorer 16 Development Board

Microchip
www.microchip.com

Using a **VEX Controller** for Electronic Experiments



By Daniel Ramirez

It is not generally known that a VEX Construction Set can be used as a platform to carry out various educational electronic experiments. Performing the experiments that will be described in this series of articles will help you understand the basic electronics that are used to read digital and analog inputs, and to control digital and analog devices connected to a standard VEX microcontroller. A VEX microcontroller will be the primary component used to carry out these experiments utilizing its versatile I/O ports as will be demonstrated here. The VEX microcontroller is well designed and very robust, and can stand up to harsh environments using internal protection circuitry which makes it perfect for the first experiment described here.

This project does require that some firmware be downloaded to the microcontroller using the Innovations First, Inc. (IFI), Bootloader. I will provide the firmware as a hex file so that readers who only want to build and run the project can do so without having to do any C programming. However, I will also provide C source code for those who do wish to know how the embedded firmware works and want to modify it.

What the VEX?

Why would you want to interface non-VEX components to the VEX microcontroller? Especially since sensors and motors sold by IFI conveniently plug into the digital and analog I/O ports. There are some sensors that IFI currently does not sell, like temperature and humidity sensors, and GPS sensors. These units might require a high speed serial protocol such as SPI or I^{2}C that is currently not supported by the standard VEX chip since it is already using the SPI peripheral internally.

I have found other instances where I needed to drive a stepper motor and high speed DC motor. A solution to this dilemma is to build your own circuits that read the needed sensors and drive the external motors. I will be showing you simple experiments that allow you to interface other kinds of non-VEX electronic components including numeric LED displays, LCD displays, keypads, temperature sensors, DC motors, and stepper motors.

The experience gained by understanding just what controllers can do and what their limitations are is a big advantage both for learning and in contest environments. Remember, modifications to the VEX Microcontroller and

interfacing your own circuits to it is usually not allowed by contest rules in VEX VRC contests. For instance, even though rules specify that only VEX sensors and actuators can be used in a contest, by knowing the internal details of the PIC18F8520 (in this case) the contestant can take advantage of using other PIC peripherals such as internal PWM timers and remapping the five interrupt pins to be used as a high speed digital I/O when programming firmware using the C programming language. In order to gain a better understanding of the Microchip PIC architecture, you will need to reference the PIC18F8520 datasheets found at **www.microchip.com**.

Some Safety Stuff

When you are connecting your own external circuits to the VEX microcontroller pins for the experiments in this article, make sure you do not directly short the power pin (middle pin) to the ground or signal pins. Otherwise, you might cause the controller to reset or overheat, making it thermally shut down or possibly even damage the VEX controller. Do not try to drive high current devices including two wire DC motors or light bulbs from the three wire PWM motor outputs. If you do need to drive two wire motors, consider purchasing the low cost VEX PWM adapter cables sold by IFI. Use a voltmeter to test the circuit if you have doubts about which ones are the power and ground pins on the VEX. During my experimentation with the VEX controller, I accidently made all these mistakes, but have not damaged it — which is a testament to just how durable it is. Whenever you are first testing a new circuit, be sure to look for any of these problems, and watch the controller's behavior; even monitor its temperature (it should be slightly warm but never hot!).

The VEX Inventor's manual does an excellent job of describing the various subsystems that make up a robot model or animation prop including: a logic subsystem; a control subsystem; and a structure subsystem. You can find it at **www.vexforum.com/wiki/index.php/Inventor%27s_Guide**.

Let's Use Some Logic

We will mainly be concerned with the VEX's logic subsystem. The front of the controller has three jacks that look like RJ 11 telephone jacks in as one power socket; to the right of the power socket is the power switch.

A VEX microcontroller provides a regulated 5 volt power supply that can also be used for these experiments, as long as the circuit power requirements do not exceed the IFI specifications of +5 volts, one amp maximum

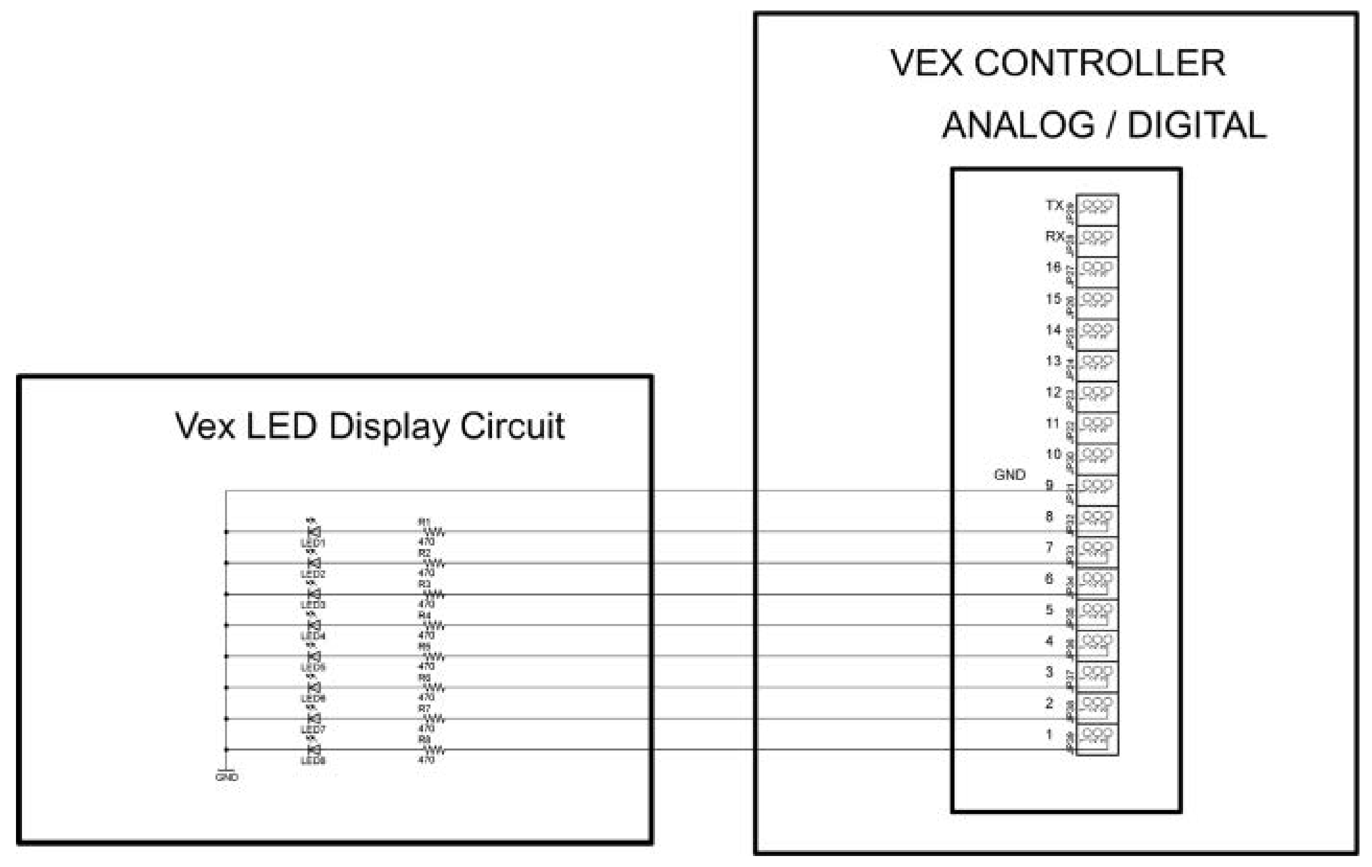


FIGURE 1. This schematic shows how to wire a simple LED circuit to the VEX microcontroller's digital output pins.

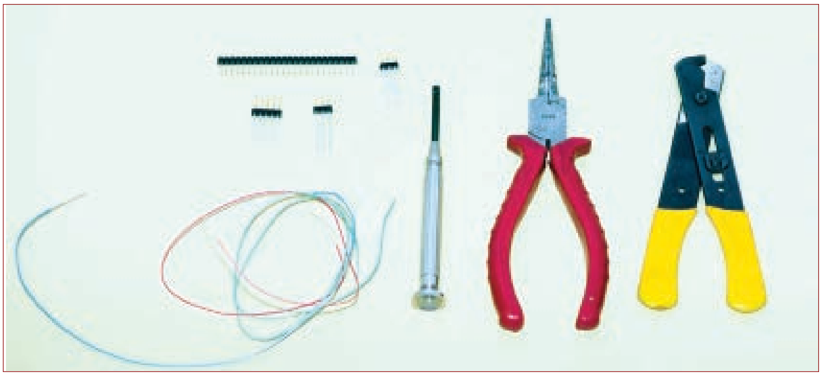


FIGURE 2. These are the tools I highly recommend for rapid prototyping of electronic circuits with the VEX microcontroller. They include wire cutters, long nose pliers, a RadioShack wire-wrap tool, and wire-wrap wire.

FIGURE 3. The "brain" of the robot. The microcontroller contains the robot's programs and processes all signals received from both human operators and onboard sensor systems. It also manages power allocation on board the robot, and directly controls the motors.

combined total from all the pins. Although these I/O pins are internally protected with resistors and diodes, they can still be damaged or shorted. Care should be taken when connecting the wires so as not to create direct shorts to ground *unless specifically directed to do so*. We must not short these pins or connect them to the power supply using the wrong polarity. The schematic in **Figure 1** shows you how to wire a simple LED circuit to the microcontroller's digital output pins. VEX motors are connected to the motor port. Everything's assembled on a RadioShack breadboard.

More Safety Stuff

Safety should always be first on the mind of any builder when working with electronics. Always wear safety eye glasses or goggles with side shields. As an alternative to soldering, (to avoid spatters) I highly recommend using wire-wrap and a low-cost tool. If you need to solder, only use unleaded ROHS solder and components when possible. Most of the experiments that will be presented here can be carried out without soldering. The tools I highly recommend for rapid prototyping are shown in **Figure 2**. The wire-wrap tool in particular is very useful for making the connections between pin headers inserted into the microcontroller's I/O ports and external circuits.

Another precaution to take is to make sure the workspace is static free in order to prevent Electrostatic Discharge (ESD) damage to the microcontroller and any circuits. Purchasing a low-cost ground strap is also a good idea.

Getting Better Aquainted

The VEX microcontroller shown in **Figure 3** is more than just a motor controller since it provides access for up to 16 digital or analog input or output pins, eight motor outputs, and six interrupts. It uses its silicon brain to process algorithms with its internal processors and memory. It will provide a convenient platform to carry out our electronic experiments and will allow us to extend the controller's input and output capabilities to read new kinds of sensors, switches, keypads, LED displays, and LCD displays.

Powering The VEX Controller

The recommended power supply to attach to the power connector (on the back of the microcontroller) is a 7.6 volt battery pack; although you can also connect a 9.6 volt battery pack since it shares the same connector and does not seem to harm the controller. Connecting the 9.6 volt battery pack does make motors and servos run faster, which could wear them out sooner. (For this reason, IFI does not recommend using the higher voltage battery pack. It also is not allowed in most VEX VRC contests.)

VEX Controller Symptoms

There are three LEDs located on the back of the controller that give you an indication of its operational state. Blinking red LEDs let you know when the battery is getting low; flashing green LEDs provide an indication that the controller is being programmed. The VEX controller may also reset when there is a direct short circuit from a power pin (+5 volts) to ground (0 volts). Be prepared to take action to protect the controller by disconnecting it from the power source (not just turning off the power switch). I have encountered some of the symptoms listed here while performing my own experiments and discovered that if you take quick actions you can avoid any permanent damage to the controller.

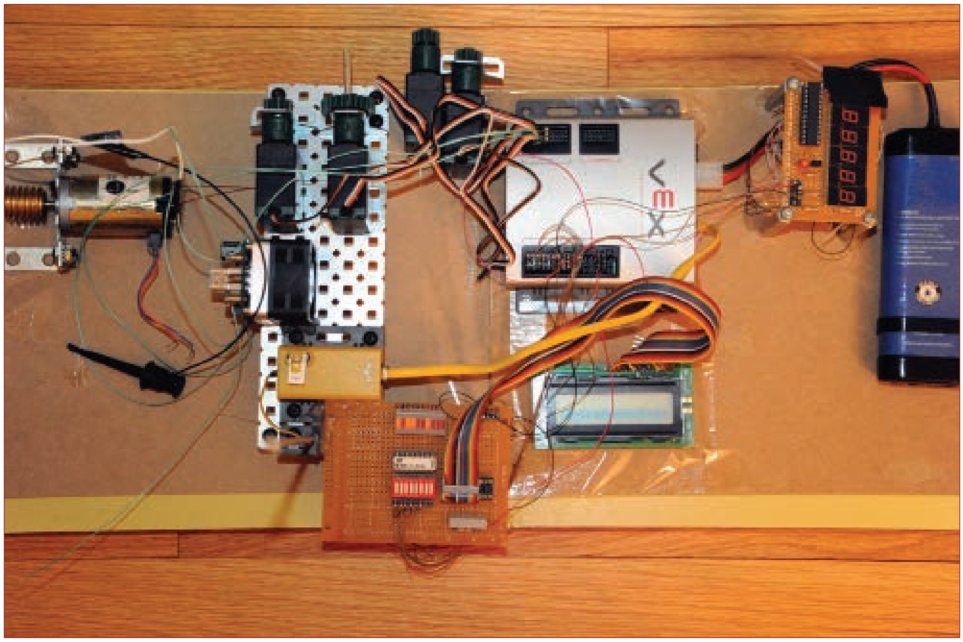


FIGURE 4. The completed circuit for the LED driver assembled on a breadboard and wired to the VEX microcontroller's digital output pins, along with the VEX 9.6 volt/1,000 mAh rechargeable battery.

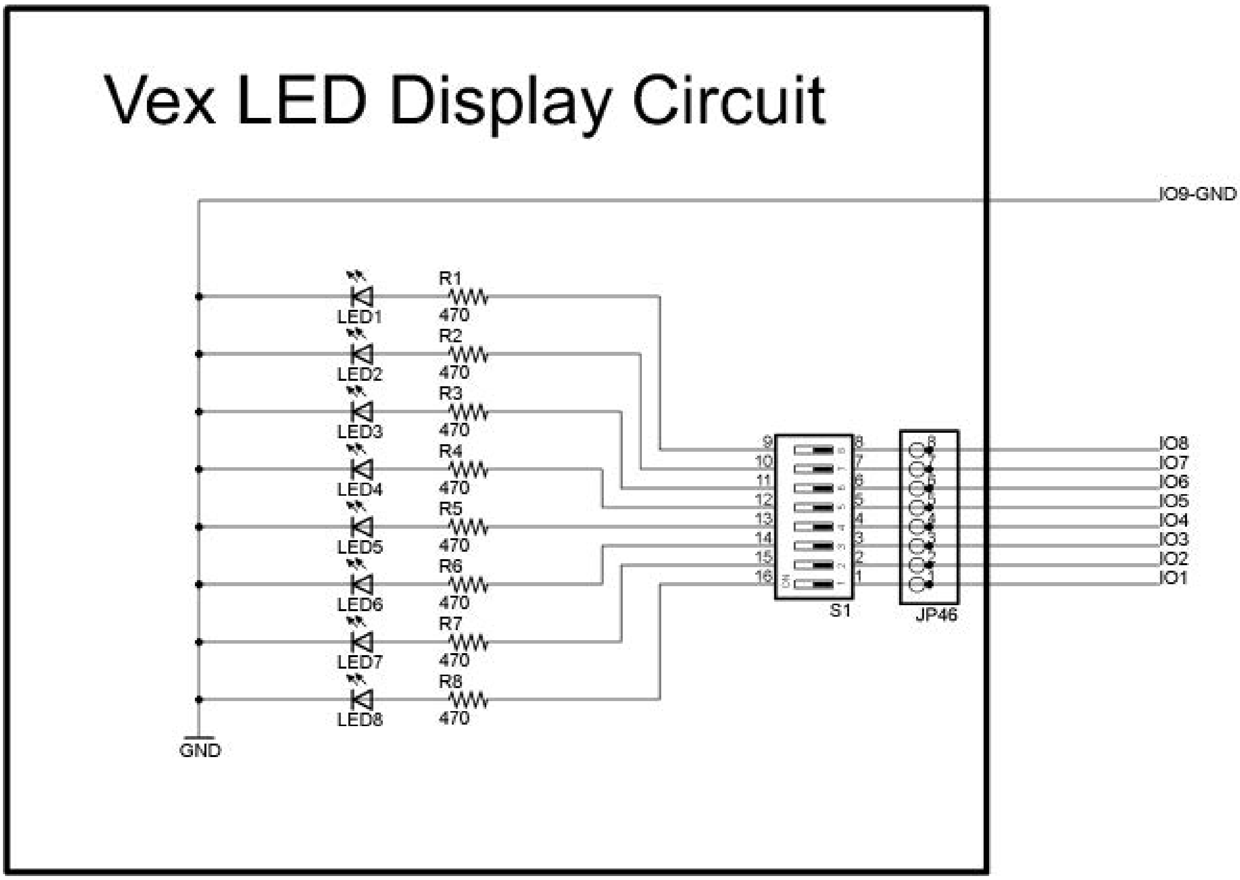


FIGURE 5. Schematic shows how up to eight LEDs can be connected to the VEX microcontroller's digital output pins. The remaining two LEDs can be used for other purposes or the display can be expanded to use all 10 LEDs. Please note that the resistors connected between the VEX controller and the LED BAR are optional.

TABLE 1. Bill of Materials (BOM) for the LED Display Board.

Item	Qty	Source	Description
1	1	RadioShack	Prototype PC board
2	1	RadioShack	10 LED BAR display
3	1	Jameco	270 ohm 16 pin DIP resistor, isolated
4	8	Jameco	Straight .001 pin headers
5	1	IFI	VEX controller

• Signs of smoke or sparks: Disconnect the controller and circuits from the power source.

• Microcontroller keeps resetting: Possible low battery voltage condition if LEDs are flashing, or could also indicate a short circuit between the power and ground pins.

• Does not download an application (hex file): This symptom could be an indication of a short to ground. Check all circuit connections for shorts using a DVM. If no short is detected, remove any external circuitry and try to download the application again after a few minutes.

• Controller feels hot to the touch: This is an indication of a short circuit. Disconnect power and remove any external circuitry connected to the digital or analog I/O pins.

• Controller resets periodically: The programming light flashing red indicates a problem with the user code. If you have a programming kit, download the default code into the Controller. Otherwise, you may need to return the unit to IFI.

The LED Display Experiment

The simplest circuit that you can build is to just insert one or more LEDs into any of the 16 digital outputs, making sure that the cathode (end with a notch) of each LED is connected to ground (-) and the anode (+) is connected to the middle port. A simple C application sets the selected output pin to 1 to turn the LED on 0 to turn the LED off. There is no need to use a current-limiting resistor since the VEX microcontroller provides a 1K ohm resistor in series with the output port.

In this experiment, the VEX microcontroller is used to blink from one to eight LEDs that are contained in a display. This will demonstrate the microcontroller's digital output capabilities. The completed circuit for the LED driver assembled on a breadboard and wired to the microcontroller's digital output pins is shown in **Figure 4**. Each LED is connected to one of the digital output pins and is toggled so that they turn on or off using a simple Easy C or PIC18 C program (as shown in **Listing 1**). In order to access the digital output pins, insert eight .100 3 x 1 pin headers into the digital/analog block on top of the microcontroller. The outputs of these pins can be connected to corresponding pin headers on a prototype board; use wire wrapper or jumper cables to make the connections as shown in the schematic in **Figure 5**. (It shows how the eight LEDs can be connected to the digital output pins.) The simple, experimental platform shown in **Figure 4** consists of the VEX microcontroller, LED display, numeric LED display, LCD display, five motors, a Parallax motor controller, and a VEX 9.6 volt battery. This setup provides the test-bed to carry out this and future experiments.

There are basically two ways to connect LEDs to microcontrollers. The first way is to source the current to the LED. This means that sending a logic 1 to an output port, representing a voltage of +5 volts (TTL) will light the LED. The second way is to sink the current to the LED by sending a 0 to the output port, representing a voltage of 0V. The **schematic** here only shows the source connection. The 9.6 volt/1,000 mAh rechargeable battery is a portable power source and is regulated internally by the microcontroller to +5 volts. This is enough to power our eight LEDs.

```
 // Setup timers, configure ports and
                     initialize variables
SetupTimers();

// Initialize the LED Display
InitializeLED();

// Test LEDs
while(1)
{
    for (i=0; i<256; i++)
    {
        // Set the LEDs connected to
        // PORTA pins
        PORTA = i;
    pause(5000);
    }

}
```

LISTING 1.

Making the LED Display

The components that I used to build the LED display circuit are shown in **Table 1**. Other than the VEX microcontroller, the required resistors and LEDs are low-cost components you can find at places like Jameco.

Port A bits 0:5 are mapped to the microcontroller's digital/analog pins 1 through 16 as shown in **Table 2**. The 16 I/O pins are mapped to the PIC18F8520 port bits also shown in the **table**). The ones used for the LEDs are noted. Other experiments will use this mapping as well.

The LED application written in Easy C counts from 0 to 255 using the eight LEDs with a .5 second pause delay between each count.

Notice how Easy C provides the necessary I/O functions to toggle the LEDs. The LEDs can also be used as a low-cost LED digital scope, logic analyzer, or logic probe.

Accessing the VEX Microcontroller I/O Ports

Now that we know that the standard VEX microcontroller uses two PIC18F8520 microcontrollers, the question is which of the external controller I/O pins correspond to the internal PIC18F8520 pins. To answer this question, refer to **Table 3**. IFI uses different names (aliases) for the ports than those specified in the PIC18F8520 datasheet. This can be a source of confusion since the PIC18F8520 port registers PORTA through PORTH are mapped to the VEX 16 digital and analog I/O, eight PWM ports, and five interrupts. **Table 3** is the Rosetta stone for being able to carry out advanced VEX microcontroller experiments by allowing you to use the internal PIC peripherals and registers directly in your application. Another resource available from Microchip is their application notes specific to the PIC18F8520. These notes provide details on using the PIC timers, PWM, and PIC peripherals including the USART, I²C, SPI, and interrupts. You may also want to learn the PIC assembly language which gives you access to almost all the registers and peripherals, and also allows you to write very fast code needed for Interrupt Service Routines (ISRs). Assembly language routines can also be called from C (mixed mode programming).

IFI aliases for the direction register are defined as follows:

- Pin directions IO1 to IO16.
- Inputs as rc_dig_in01 to rc_dig_in16.
- Outputs as rc_dig_out01 to rc_dig_out16.

The code to turn the LEDs on or off is leds.c which is available in the downloads for this article at **www.servomagazine.com**. It was developed using MPLAB and the PIC18 C compiler. It configures the VEX controller's digital ports to outputs and sets each LED segment to the desired state. In the code, you see an infinite while loop and an inner loop that toggles the LEDs in binary from 0 to 255. Once compiled, the leds.hex file is generated that is used to program the VEX controller. The leds.hex file is provided as well so that it can be used without having to compile it.

You'll need programming cable to download the necessary firmware.

Programming the VEX Controller

To program the microcontroller for semi-autonomous or fully autonomous operation, there are a few programming tool options which include Easy C 2.0, Easy C Professional, ROBOT C, and PIC18 C with MPLAB. The firmware code generated by these compilers can be downloaded using the IFI bootloader or the loader application that comes with each product. Firmware files have the extension filename.hex; they are not easy to read since they are in hexadecimal format.

VEX PIC18F8520			
PIN	**DIRECTION**	**INPUT**	**OUTPUT**
1	TRISAbits.TRISA0	PORTAbits.RA0	LATAbits.LATA0
2	TRISAbits.TRISA1	PORTAbits.RA1	LATAbits.LATA1
3	TRISAbits.TRISA2	PORTAbits.RA2	LATAbits.LATA2
4	TRISAbits.TRISA3	PORTAbits.RA3	LATAbits.LATA3
5	TRISAbits.TRISA5	PORTAbits.RA5	LATAbits.LATA5
6	TRISFbits.TRISF0	PORTFbits.RF0	LATFbits.LATF0
7	TRISFbits.TRISF1	PORTFbits.RF1	LATFbits.LATF1
8	TRISFbits.TRISF2	PORTFbits.RF2	LATFbits.LATF2
9	TRISFbits.TRISF3	PORTFbits.RF3	LATFbits.LATF3
10	TRISFbits.TRISF4	PORTFbits.RF4	LATFbits.LATF4
11	TRISFbits.TRISF5	PORTFbits.RF5	LATFbits.LATF5
12	TRISFbits.TRISF6	PORTFbits.RF5	LATFbits.LATF6
13	TRISHbits.TRISH4	PORTHbits.RH4	LATHbits.LATH4
14	TRISHbits.TRISH5	PORTHbits.RH5	LATHbits.LATH5
15	TRISHbits.TRISH6	PORTHbits.RH6	LATHbits.LATH6
16	TRISHbits.TRISH7	PORTHbits.RH7	LATHbits.LATH7

TABLE 2. This table is the Rosetta stone for being able to carry out advanced VEX microcontroller experiments since it shows the mapping of the PIC18F8520 ports to the controller's digital input/output pins located on the block on top of the VEX controller. The firmware defines aliases for each of the PIC18F4520 port registers in the ifi_aliases.h file.

The next step is to download the provided led.hex application into the controller and run it. Start by copying it to your laptop or PC hard disk and place it in a folder.

Once the led.hex file has been downloaded to the VEX; the LEDs should start flashing in a pattern.

That's a Wrap

This experiment — while relatively simple — can be used in VEX-based projects and provide status indicators using discrete LEDs, display voltage levels and motor speed using the LED BAR, and even display eight to 10-bit numeric quantities in binary. I hope you've gained insight in how the VEX digital and analog outputs are mapped.

See you "vex" time. **SV**

By Chris Savage

FIGURE 1. Big Trak (circa 1980).

GPS Navigation Part 1

In the 80s when I first started building robots, I found there were roughly three ways I could program my bot for autonomous navigation depending on the sensors. The first option was basic roaming where the robot would simply move in a given direction until a sensor detected something in its path, at which point the robot would choose a new path (based on where the object was detected). The second option was to use beacons or a line to guide the robot from one point to another.

Finally, a third method known as dead reckoning was demonstrated to me via a plastic toy tank created by Milton Bradley in 1979 (see **Figure 1**). This six-wheeled toy tank known as "Big Trak" was probably the most advanced *toy* I had seen in awhile. It used an internal optical wheel encoder to determine how far it moved and could execute commands telling it how far to go in a direction, how much to turn, and even to stop for periods of time.

Wide Open Spaces

In a living room or kitchen environment, the Big Trak could be placed in a fixed spot at a fixed angle and expect to navigate using dead reckoning with fairly good repeatability. However, the playing field opened up for vehicles that can move faster and farther such as radio controlled boats, planes, and even larger robots. In these situations, dead reckoning really isn't that useful. To complicate things even more, most typical sensors don't have the range to handle avoidance at the speeds these vehicles move at. Lines become useless and beacons

become impractical. In a larger arena, you really need something that works on a much larger scale.

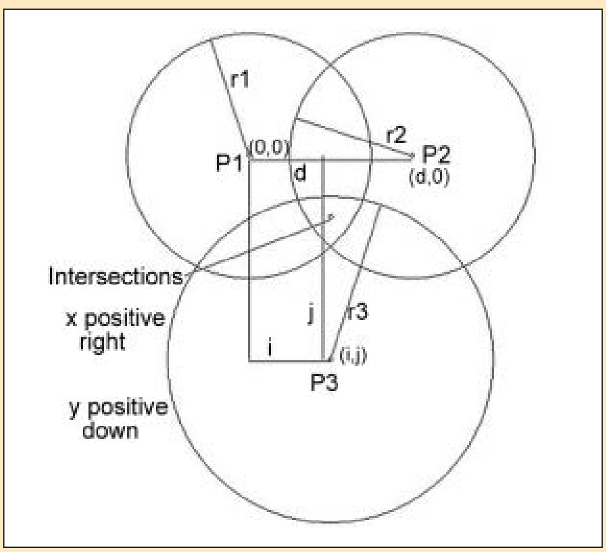
Fast Forward

What wasn't available to the general public 20 years ago has now become widely available and extremely accessible to the robotics hobbyist. GPS — or the Global Positioning System — can provide larger vehicles and robots a new method of navigation. GPS receivers are integrated into cell-phones, vehicles, and even camcorders now. For the hobbyist, GPS has become widely available from sources such as Parallax, Inc. (**www.parallax.com**), for well under $100 (see **Figure 2**). Using GPS, a robot, boat, or plane can easily travel great distances autonomously with very little extra hardware.

How Does It Work?

GPS works by calculating a position based on signals received from a network of satellites in orbit around the earth. These satellites send their time of transmission,

Trilateration is a method for determining the intersections of three sphere surfaces given the centers and radii of the three spheres.



The plane, z=0, showing the 3 sphere centers, P1, P2, and P3; their x,y coordinates; and the three sphere radii, r1, r2, and r3. The two intersections of the three sphere surfaces are directly in front and directly behind the point designated intersections in the z=0 plane.

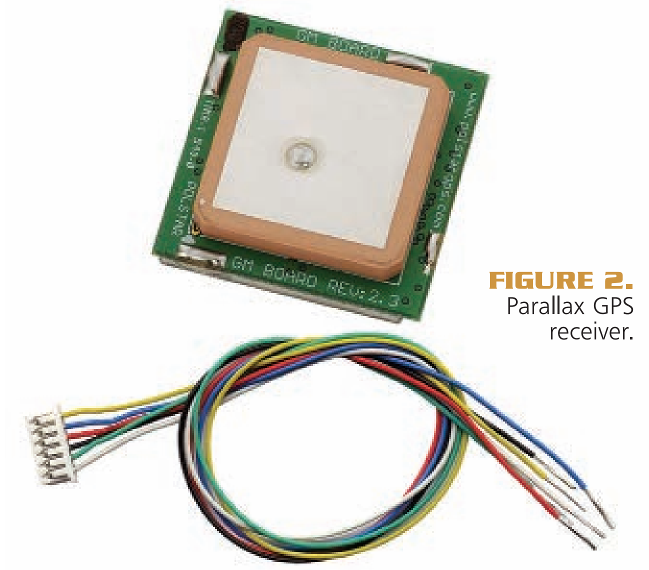


FIGURE 2. Parallax GPS receiver.

precise orbital information, and the signal strength and health of the satellites. The GPS receiver uses this information to calculate its position on the globe using trilateration. Like dead reckoning, GPS is prone to errors and a small error can cause a big change. At least three satellites are required for the GPS to have valid data with which to calculate positions; more satellites mean more accuracy.

Bringing It All Down To Earth

If you really want to understand trilateration, there's an excellent article on Wikipedia that explains how GPS receivers do this. For our purposes, we will be dealing with a flat map and a much smaller area in terms of navigation. The GPS receiver will provide us with several key pieces of information such as latitude, longitude, altitude, and time. Most also provide heading and speed, however, I recommend only using the latitude, longitude, and time. Altitude on most GPS receivers is not nearly as accurate or consistent as position. Heading and speed are derived from changes in position and are also not very reliable. Also, since you have to be moving to get either value they're useless when not moving.

Sensor Fusion

Adding a compass module such as the HMC6352 to your GPS receiver allows you to get a more accurate heading. This compass (see **Figure 3**) can be obtained from Parallax, as well. The first steps in fusing our data from the GPS and the compass module start by knowing the destination coordinates, as

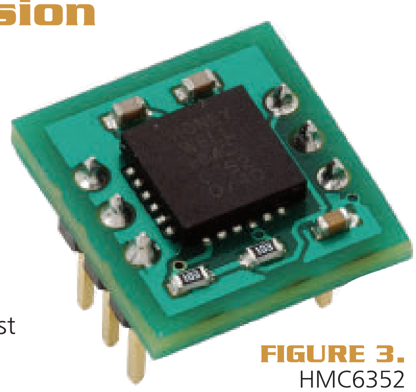
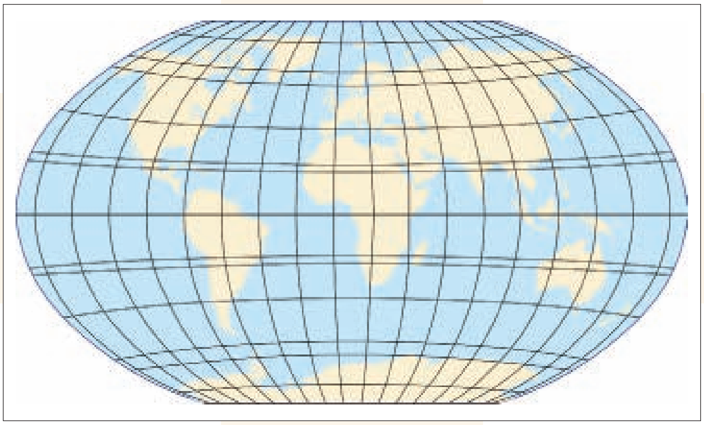


FIGURE 3. HMC6352 compass module.

FIGURE 4. Simple map of the earth showing latitude/longitude lines.



well as the current coordinates. The destination coordinates are often known in advance and programmed into the robot ahead of time.

This same concept is the basis for the RoboMagellan competition where robots are given access to the GPS coordinates of the waypoints on the course, and the robot must navigate from its current position to the point or points on the map. Each waypoint is designated by a large orange traffic cone allowing the robot to confirm it has reached that waypoint visually.

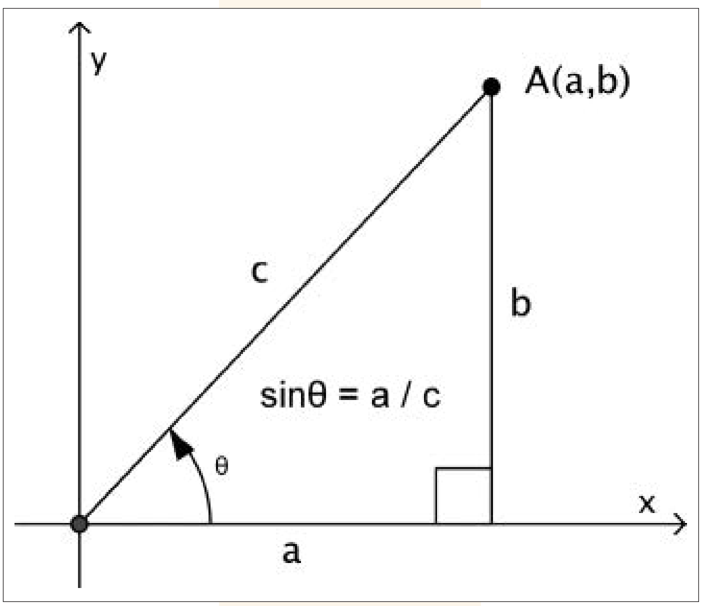
Going Back To School

In order to calculate the heading to the next waypoint from the current position, you have to do a little math. To simplify things, let's assume we won't be crossing the equator or the prime meridian since this is unlikely anyway. By doing this, we can treat all latitude and longitude values as absolute. So, to calculate distance and heading you will plot a point at the coordinates you are heading to relative to the coordinates you are at. Looking at **Figure 5**, we will say that the point where **x** and **y** intersect is our home point. This represents the current coordinates. Point **A** represents the point where our waypoint destination is.

Essentially, what we've done is created a virtual right triangle which will allow us to get two important values. By treating all values as absolute, you can simply subtract the **x** values to get **a** and subtract the **y** values to get **b**. Using the Pythagorean Theorem, you can now calculate **c**. Remember, $\mathbf{a}^2 + \mathbf{b}^2 = \mathbf{c}^2$. Now we have our distance which can be used in a robot with wheel encoders to ensure we travel far enough. Now to calculate the heading.

Some trigonometry is involved in this step. In order to obtain θ (Theta), you will take the inverse of the formulae within **Figure 5**. This will give you the degrees of the heading you need to be on. Based on your current heading, you can then calculate the new heading. For the most part, you won't have to worry about the calculations. They should all be tucked into a nice subroutine. About the only thing not covered here is some filtering to smooth out changes. Once things are dialed in, the robot shouldn't veer much. As a safety, we can make it stop whenever there is no GPS lock if need be.

FIGURE 5. Pythagorian Theorem and formulas.



Final Thoughts

The specifics of handling the data from the compass module will vary depending on which compass module you use and how its data is formatted. The specifics of dealing with the distance data will vary some as well, depending on your microcontroller, drive system, encoders, and overall resolution. In Part 2, we will explore a real world example of having a robot navigate through several waypoints to reach a destination. Based on the information provided here, I can almost see the light bulbs over your heads and the wheels turning (both figuratively and literally). You've got a month to implement these ideas. Until then, brush up on your math and have fun experimenting! **SV**

Resources

Right Triangle Trigonometry
www.tiem.utk.edu/bioed/bealsmodules/triangle.html

Project Page
www.savagecircuits.com/gpsnav/

Parallax, Inc.
www.parallax.com

Wikipedia
en.wikipedia.org

ROBOGAMES

World's Largest Robot Competition
-Guinness Book of Records

North America's Top Ten Geek Fests
-Wired Magazine

SportCenter's Top Ten
-ESPN SportsCenter

"If you are a robot enthusiast, I would definitely encourage you to attend the RoboGames... Take a plane, train, space elevator, but definitely go!"
-Servo Magazine

"Impossible to Imagine, Impossible to Forget!"
-Robot Magazine

Compete at RoboGames!
Last year, over 1000 builders from around the world brought over 800 robots to San Francisco, in the 4th annual international event. This year, we expect even more robots and engineers to compete. Be one! With 80 different events, there's a competition for everyone - combat, androids, sumo, soccer, Lego, art, micromouse, BEAM, or Tetsujin! More than half the events are autonomous. Even if you just come to watch, you'll be overwhelmed with the diversity.

Last year, RoboGames hosted teams with over 800 robots from Argentina, Australia, Austria, Brazil, Canada, China, Colombia, Czech Republic, Denmark, Germany, India, Indonesia, Iran, Japan, Korea, Mexico, Netherlands, Peru, Singapore, Slovenia, Sweden, Taiwan, UK, and the USA.

Be a RoboGames Sponsor!
RoboGames is the world's largest open robot competition - letting people of any age, gender, nationality, or affiliation compete. Sponsoring RoboGames not only helps more people to compete, but also gets your company unrivaled press coverage and visibility. The event has been covered by CNN, ESPN, Fox, CBS, ABC, NBC (live), EBS Korea, NHK Japan, BBC, and countless print and web companies. Your logo can be everywhere the cameras turn!

Rent a Booth!
Booth spaces are at the front of the venue, ensuring lots of traffic. With 3000-5000 people each day, you're company will get amazing traffic!

Events:

Combat: 340 lbs, 220, 120, 60, 30, 3, & 1 lbs

Androids: Wrestling, Demonstration, Stair Climbing, The Eagle, Door Opening, The Toss, BasketBall, Lift and Carry, Marathon, Obstacle Run, Penalty Kick, Dash, 3:3 Soccer, Weight Lifting

Open Events: Fire-Fighting, Robomagellan, Maze/MicroMouse, Walker Challenge, Biped Race, Robot Triathlon, Line Slalom, Ribbon Climber, Vex Open, Lego Challenge, Lego Open, Aibo Performer, Balancer Race, Best of Show, Bot Hockey

Sumo: 3kg - Auto & R/C, 500g , 100g , 25g, Humanoid

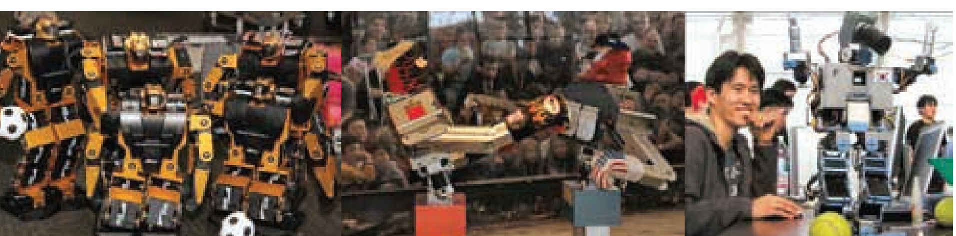
Robot Soccer: Biped 3:3 & 5:5 , Mirosot 5:5 &11:11

Junior League: Lego Challenge, Lego Open, Lego Magellan, Woots & Snarks, Handy Board Ball, BotsketBall, 500 g Sumo, 120 lb combat, Best of Show, Vex Open

Tetsujin (ExoSkeleton): Lifting, Walking, Carrying

Art Bots: Static, Kinetic, Bartending, Musical, Drawing

BEAM: Speeder, Photovore, RoboSapien Hacker



April 23-25, 2010
San Mateo, CA
http://RoboGames.Net

GET READY FOR ROBOGAMES 2010!

By Evan Ackerman

RoboGames is the largest open robot competition in the world, and the 2010 competition (taking place April 23-25 in San Mateo, CA) promises to be the most exciting, creative, and just generally incredibly fantastically awesome event yet. The first RoboGames was held seven years ago, and since then, each year has brought more people, more events, and more robots. Last year, 620 people and 403 robots came to the San Francisco Bay Area to experience three days of robot combat, robot sporting events, and robot artwork.

The most popular — and arguably the most dramatic — part of RoboGames are the robot combat events. Bots ranging from five ounces to 340 pounds attempt to destroy each other (and quite often succeed) in a giant armored arena surrounded by bulletproof glass. Some robots are simple wedges designed to flip their opponents, while others use complicated active weapons like titanium blades (spinning at over 5,000 rpm), pneumatic lifting arms, and flamethrowers. The robots are remote controlled, and after each match, the loser (and sometimes the winner) gets rushed off to the pits to repair the inevitable damage while the scent of scorched metal, fried speed controllers, and battery acid drifts through the crowd.

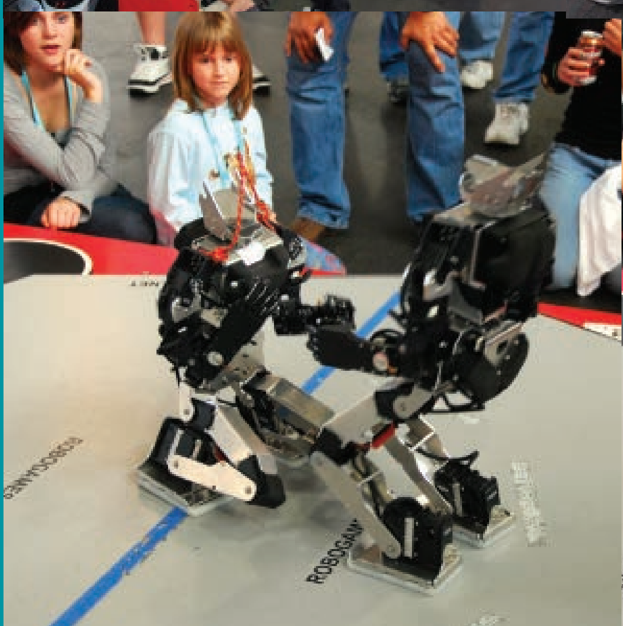
While RoboGames is perhaps best known for combat robots, robot-on-robot violence is only a small part of the games as a whole. There are over 70 different events that bots of all shapes and sizes can compete in — from maze solving to ribbon climbing to autonomous firefighting. Humanoid robot competitions include Kung-Fu, remote controlled and autonomous soccer, and autonomous basketball. And it's not just fun and games, either. Part of the idea behind RoboGames is to get roboticists from a variety of backgrounds and with many different areas of expertise in one place, where they can collaborate and learn from each other.

Entrants range in age from seven to 72, come from 20+ countries, with women comprising about a third of the participants. All too often, building robots is something that people tend do by themselves in their basements. but when there's an international competition to bring your robot to, you get to see what everybody else has been working on (in their basements). Plus there's nothing like a second place finish to motivate competitors to get even more creative with their bots.

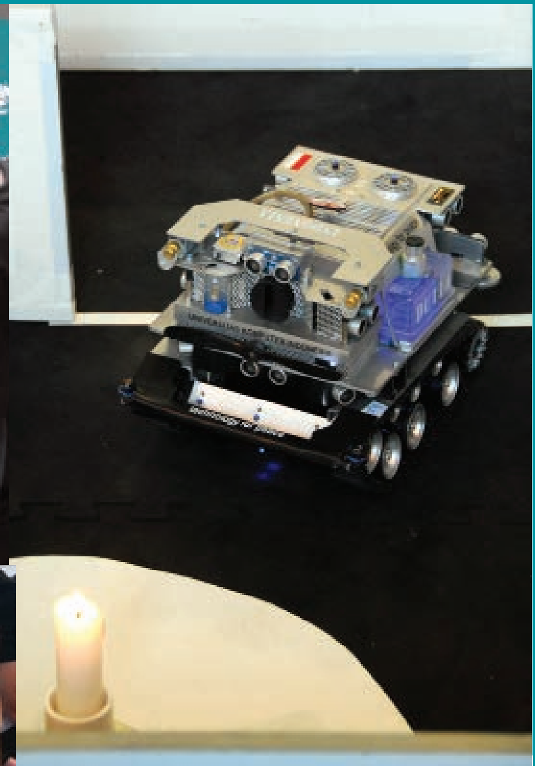
One of the most exciting events that made its debut at RoboGames last year was Mech Warfare. Mech Warfare pits biped and quadruped robots armed with airsoft weapons against each other in a 15' x 15' urban arena. The robots are controlled remotely, with their operators viewing the action only through wireless video cameras mounted on the robots which provide a first-person perspective like the classic video game. In addition to the airsoft class, there's also a 'hardcore' class which allows much more powerful weapons, such as CO_2 rifles, flamethrowers, and rockets.

As the first Mech Warfare event ever, the 2009 competition was certainly a success, with a quad mech named 'Issydunyet' (built by Michael Ferguson) taking the gold. What won the competition for Issy was its impressive mobility, and it quickly became apparent that things like payload capacity, battery life, and overall robustness and reliability were far more important than firepower. The 2010 competition is sure to be even fiercer, with all of the competitors having had a year to update their robots and their strategy.

All RoboGames events are open to the public, and the action runs from noon until 7 PM Friday, Saturday, and Sunday April 23-35 at the San Mateo County Fairgrounds in San Mateo, CA. Admission is free to kids under 17 on Friday, and $20 a day for adults with significant discounts for military personnel. For more information and to purchase tickets, visit **RoboGames.net**. SV

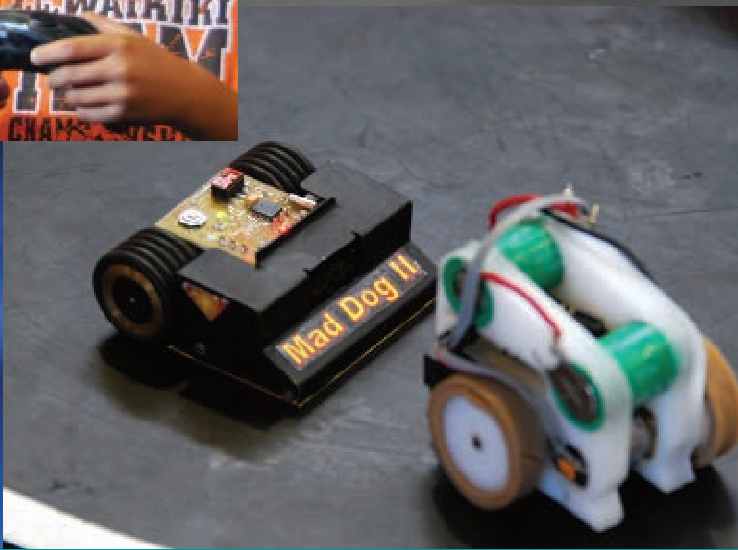


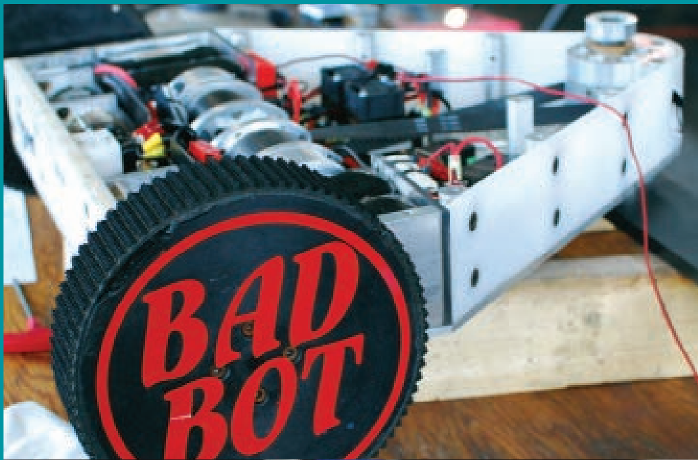
worldmags

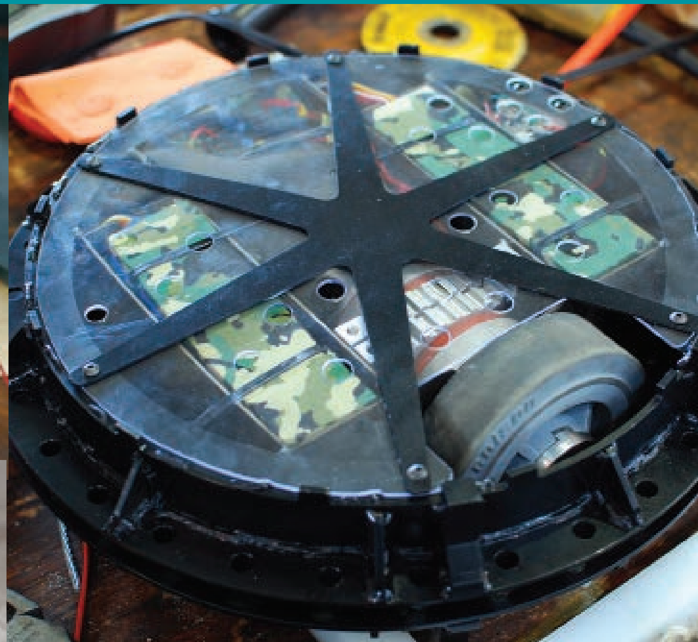


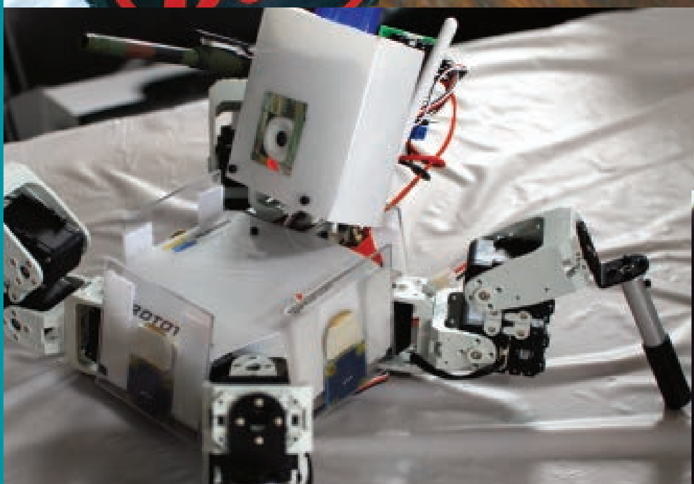

LC WAIKIKI

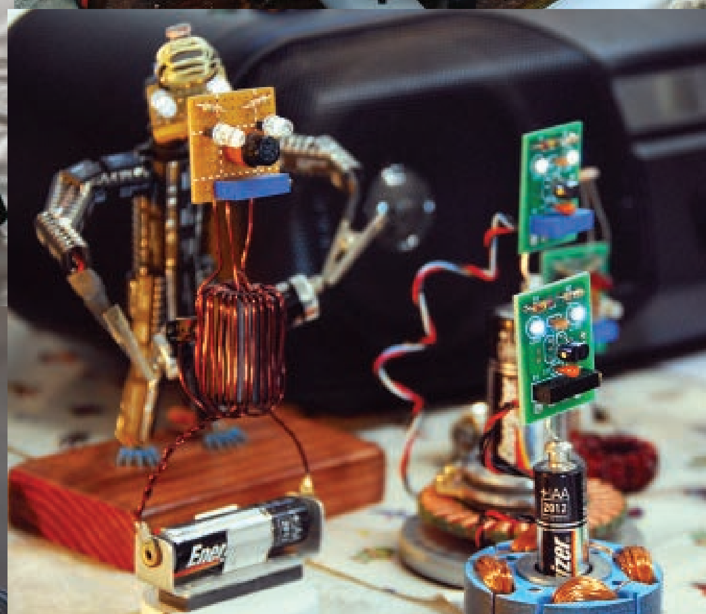

Mad Dog II

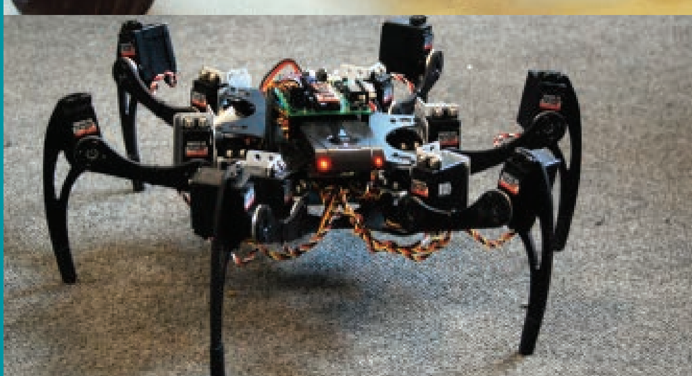

Mad Dog II


BAD
BOT

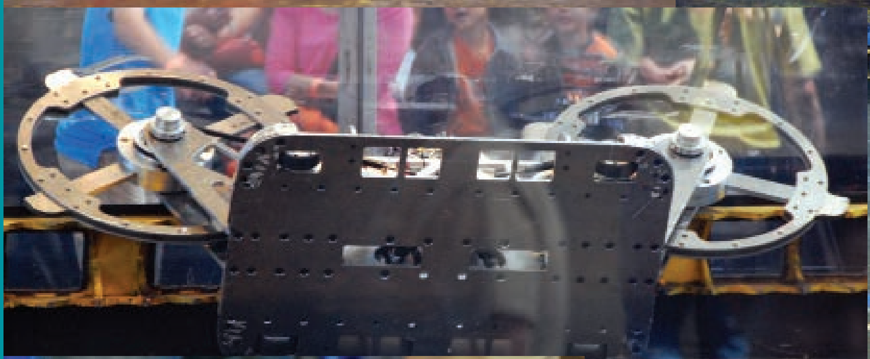















By Brennon Williams

An Introduction to Microcontrollers:

The Good, the Bad, and the Ugly

Microcontrollers are the driving component behind nearly every digital device. They control everything from the cell phone in your pocket to the NASA satellites hurling above Earth at a speed of over 17,000 miles per hour. What microcontrollers do is process inputs from the real world — such as a temperature sensor — and then respond as programmed by flipping a series of switches on or off, as with a light emitting diode (LED).

Microcontrollers are sometimes referred to as "single chip computers." This is because that's exactly what they are — entire computers sitting on a single integrated circuit. Much like your Mac or PC, they use binary code (zeroes and ones) to store data. Also like your computer, they have a hard disk of sorts. It is called Flash memory and it retains information even when the device is off.

There are other forms of memory such as Random Access Memory (RAM). However, when your microcontroller is turned off, all data stored in RAM is lost.

Regardless of memory and power, the main thing to note is that microcontrollers are designed to react to input from their environment as opposed to from a user.

Microcontrollers are often confused with microprocessors. Microprocessors are typically much higher performing, and are used in desktop and laptop computers. Unlike their heavy-duty cousin, microcontrollers have the unique ability to use incredibly small amounts of electricity in milliwatts or microwatts. Many microcontrollers can go into a sleep mode (such as when waiting for someone to push a button) that can bring the power usage down to nanowatts, making them invaluable for devices that need to have a long battery life. Microcontrollers can also have clock rates as low as 4 kHz. By contrast, my Apple MacBook operates at 2.16 GHz, dwarfing the microcontroller in comparison.

While microcontrollers can operate at extremely low speeds, they can also be used for more demanding tasks like taking in and logging GPS coordinates in real time, or mapping out the surrounding terrain of a robot using laser scanners.

For engineers, the microcontroller is an important part in nearly all of their projects. Picking your first microcontroller is like choosing your first car. There are a lot of choices and multiple things to consider that relate to each job requirement. I'll walk you through three of the more popular microcontrollers, and give you some ideas on how they might be used and how to avoid common pitfalls.

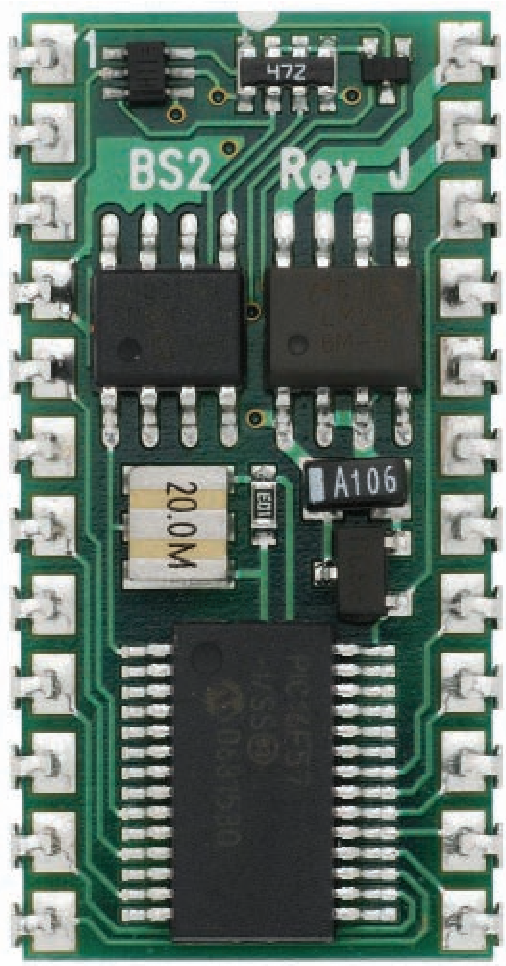
I remember buying my first microcontroller like it was only yesterday. I purchased the "What's a Microcontroller?" kit from RadioShack for about $80. As I handed it to the

guy behind the counter, he gave it a surprised look as if to say, "Wow, someone actually bought this thing!" After setting up the board and software, I stepped through a series of projects ranging from blinking LEDs to timer games.

While it was a great experience, I wasted time learning about a microcontroller I would never use again, namely because it was overpriced and impractical. Three years later, I sold it on eBay. When I did I thought to myself, "Wow, someone actually bought this thing!"

Hopefully, I'll help you avoid some of the mistakes I made when learning about microcontrollers and help you figure out which one is right for you.

BASIC Stamp II



Let's start off with the BASIC Stamp II. The BS2 has some decent technical specs. Its processor runs at 20 MHz; this is fairly fast and will work with most all small to medium sized projects you undertake. It has 32 bytes of RAM, and executes about 4,000 PBasic instructions per second (more about this in a bit). It also has 16 input/output pins. This is really important since the number of pins determines how much hardware you can hook up to it. Sixteen pins will work most of the time for projects like robots and weather stations, but if you're building something that takes data from a GPS module, an array of sensors, or a variety of lights and buzzers, I suggest you demand more connectivity.

The BASIC Stamp is coded in a form of Basic called PBasic. I was really pleased with PBasic because it was the first hardware-focused language I learned. Usually when you mention Basic to someone, they'll either make a sour face or a nostalgic one. While Basic is a very old language, it's fantastic for hardware. This is because the syntax is so easy and, well, basic (pun intended). PBasic is equally simple and is designed for the BASIC Stamp so you do not have to do anything special when writing programs for your chip (like importing separate libraries or something equally frustrating).

This may sound funny, but another consideration for using the BS2 is the great user manual. It is an entire book full of projects ranging from simple to advanced.

However, as with any product, there are naysayers who are strongly opposed to the BASIC Stamp family. When I first started with robotics at age 12, I thought I'd be using the BS2 for all my projects. I posted a question on a robotics forum asking for help hooking up a sensor, and wound up getting a lot of negative responses. Having a large support group for the microcontroller you're using is really important. Eventually, if not almost immediately, you will have something go wrong and will need someone to hold your hand and take you through all the steps to fix it.

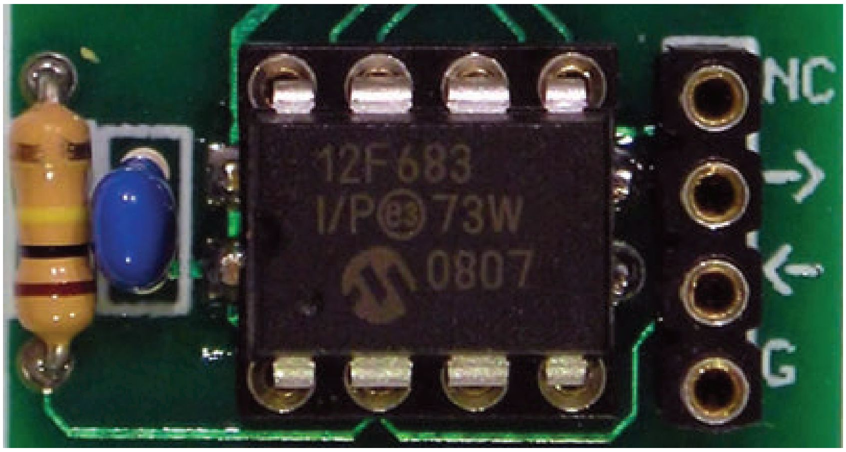
Parallax, Inc. (the manufacturer), is often criticized in the blogosphere for overpricing their products. The BASIC Stamp II really isn't a bad microcontroller, but many feel its price makes it a lower value against newer ones that are now available. Nearly all of its features and specs can be matched (if not beat) by other microcontrollers that cost $30 or less.

Let's recap the BASIC Stamp:

Pros:
- Great manual for beginners.
- Easy to learn programming language.

Cons:
- Relatively expensive

PICAXE



Next up is the PICAXE microcontroller. The PICAXE is made in the United Kingdom and a lot of its users are based there. This is of little consequence though, because you can buy them from US-based vendors (like SparkFun) and support forums on the web consist of people from all over.

What I love about the PICAXE is the wide variety of chips it can contain — everything from the tiny eight-pin to the mighty 40-pin units. Regardless of the scale of your next project, chances are there's a PICAXE to fit your needs. There are several kinds of boards, as well. There is a small eight-pin prototyping board which costs about $4. On the other hand, there are some really large boards for mega projects, like the 40-pin style which is about $30.

The smallest PICAXE is the 08, which was later replaced by its superior cousin, the 08M. An 08 sells for about $3, making it ideal for smaller projects that don't require as much code and horsepower. The 08 can support about 40 lines of code, so you will need to trim down your scripts to fit. The PICAXE team is clever since they design many of their boards with hobbyists like you and me in mind. For instance, their Axe023 board has a built-in H-bridge 1A motor driver (the microcontroller can control motors for things like robots), so making a device that can putter around is really simple.

Another great thing about the PICAXE is that all you have to pay for is the hardware. Once you have your chip, all the software and documentation is included. PICAXE has three manuals in .pdf format on their website, and each one provides a wealth of information — more than enough to educate any user on the intracacies of the PICAXE development tool. The software editor is straightforward and easy to use. Just type the code and click "upload" to load it onto the chip that's connected to your Mac or PC.

All the code for PICAXE is written in Basic, and is not hard to learn with practice and the available manuals. I find Basic to be the ideal language for hardware; it's a breeze compared to other languages like C. PICAXE code is pretty straightforward. You still use all the things other languages have (like loops), but you can do it with a lot less effort. About three lines of code can give life to any hardware you have connected to your chip.

There is a very active user group for the PICAXE. On their forum (**picaxeforum.co.uk**), users share ideas and help each other out on a daily basis; this gives the beginner the ideal support he or she needs.

My PICAXE recap:

Pros:

- Relatively low cost, wide variety of chips to choose from, good programming language, easy to use, large support group.

Cons:

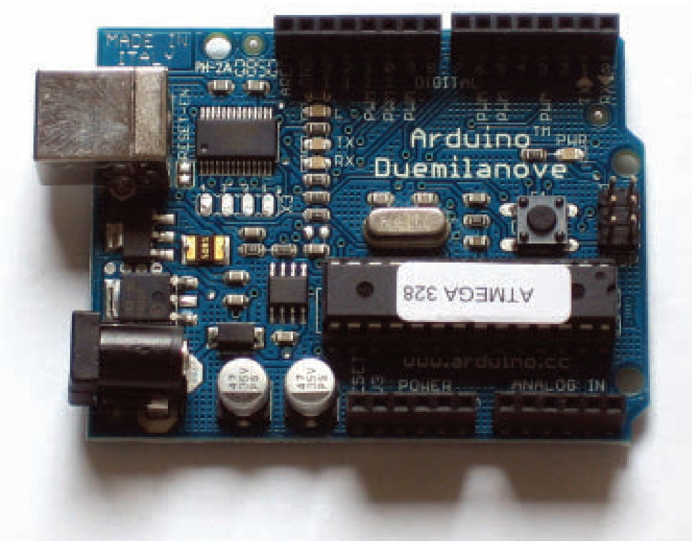
- From an aesthetics perspective, PICAXE has some rather ugly boards (orange and black are so last year).

Arduino

Our last microcontroller to look at is the loveable little blue board that has scaled the microcontroller ranks to find a special place in the hearts of thousands worldwide: the Arduino. The Arduino differs from the PICAXE and the BASIC Stamp families quite a bit. In fact, it runs off a completely different chip. Both the BS and PICAXE run on a PIC chip. The Arduino is powered by an ATMega.

Only five years ago, the Ardunio was familiar to only a few and had a very small group of users. Today, the Ardunio is one of the most popular microcontrollers in the hobbyist world.

Despite its power, the Arduino is great for the beginner as there is a nearly unlimited amount of documentation. Several books have been written focused on Arduino and the Internet is populated by hundreds of free tutorials. "Getting Started with Arduino," by Massimo Banzi is one of the better beginner microcontroller books I have ever read. Even the least tech-savy user will be capable of building



simple devices after reading this book.

The Arduino USB board uses an ATMega328. This is a powerful chip processor that runs at 20 MHz. This is the same speed as the BASIC Stamp II module we talked about earlier and is only a fraction of the cost. It can hold more lines of code than you'll need in almost all cases, and provides a high level of efficiency for a very reasonable price.

There are a few choices with the Arduino and, depending on the size of your project, you'll need to make some tough decisions. The most popular is the Arduino Duemilanove (the one that has the ATMega328) and it costs about $30. Power users will want to consider the Arduino Mega with more I/0 pins for about $65. There is also the Arduino mini (about $20), and the Arduino nano ($30-$60).

You can code the Arduino in a variety of languages, though one of the most popular is Processing; it is based on Java. Processing is easy to interface with hardware and is no harder to learn than Basic (many think it's easier).

One thing that sets the Arduino apart from other microcontrollers is the wide variety of shields available. Shields snap on top of an Ardunio (usually Duemilanove) to add hardware easily. There's a prototyping shield available, which is essentially a breadboard that makes connecting hardware very easy. There's also a GPS shield, a Bluetooth shield, an Ethernet shield, and many others. Oftentimes, buying a shield will be an easier solution than trying to figure out which wire goes in which pin.

The Arduino recap:

Pros:

- Low cost, very popular, lots of documentation, easy to learn, good programming language, variety of shields available.

Cons:

- There really aren't low cost boards like the PICAXE $4 protoboard. However, this is not a problem since most projects will call for a larger chip than an eight-pin variety.

Summary and Recommendation

Now that we've covered three of the main microcontrollers, here's what I would recommend. For small projects and to just get started in basic microcontroller usage, use the PICAXE 08 or 18-pin. For simple robots, use a PICAXE board with the built-in motor driver. For all other projects — especially those that are more complex or someday will need more capability and flexibility — give strong consideration to the Arduino family of products. SV

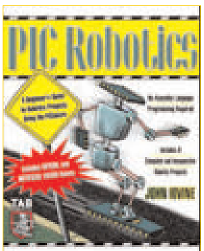
The SERVO Webstore

Attention Subscribers ask about your discount on prices marked with an *

CD-ROM SPECIALS

ROBOTICS

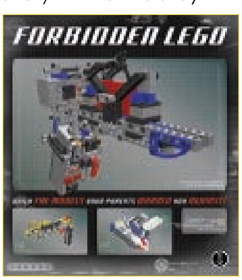
PIC Robotics
by John Iovine



Here's everything the robotics hobbyist needs to harness the power of the PICMicro MCU!

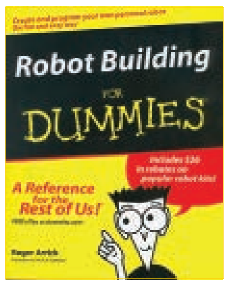
In this heavily-illustrated resource, author John Iovine provides plans and complete parts lists for 11 easy-to-build robots each with a PICMicro "brain." The expertly written coverage of the PIC Basic Computer makes programming a snap – and lots of fun.
$24.95

Forbidden LEGO
by Ulrik Pilegaard / Mike Dooley



Build the Models Your Parents Warned You Against.
Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse. **$24.95**

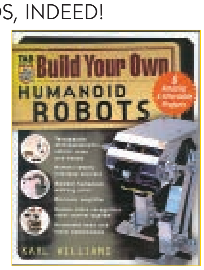
Robot Building for Dummies
by Roger Arrick / Nancy Stevenson



Discover what robots can do and how they work. Find out how to build your own robot and program it to perform tasks. Ready to enter the robot world? This book is your passport! It walks you through building your very own little metal assistant from a kit, dressing it up, giving it a brain, programming it to do things, even making it talk. Along the way, you'll gather some tidbits about robot history, enthusiasts' groups, and more.
$24.95

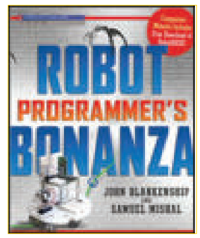
Build Your Own Humanoid Robots
by Karl Williams

GREAT 'DROIDS, INDEED!



This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot. **$24.95***

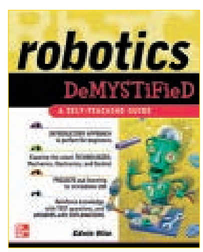
Robot Programmer's Bonanza
by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.
$29.95

Robotics Demystified
by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!



Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles

Published by T & L Publications, Inc.

$57 per bundle

Save $10 off the normal price!!

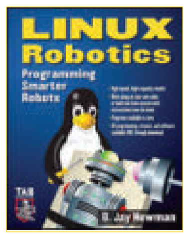
Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, and 08.

Kickin' Bot
by Grant Imahara

Enter the arena of the metal gladiators!

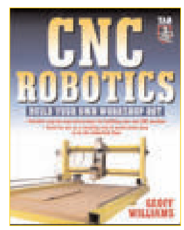
Do you have what it takes to build a battle-ready robot? You do now! Here are the plans, step-by-step directions, and expert advice that will put you in competition — while you have a heck of a lot of fun getting there. Grant Imahara, the creator of the popular BattleBot Deadblow, shares everything he's learned about robot design, tools, and techniques for metal working and the parts you need and where to get them.
$24.95

Linux Robotics
by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.
$34.95

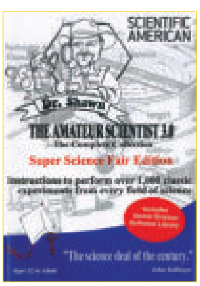
CNC Robotics
by Geoff Williams



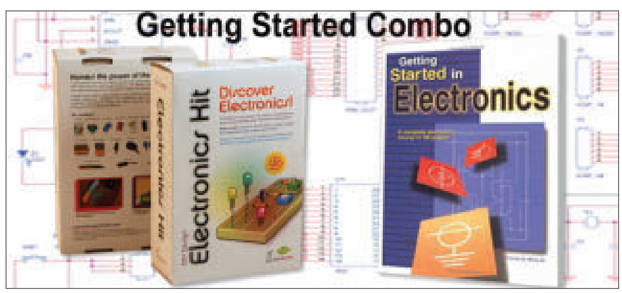
Here's the FIRST book to offer step-by-step guidelines that walk the reader through the entire process of building a CNC (Computer Numerical Control) machine from start to finish. Using inexpensive, off-the-shelf parts, readers can build CNC machines with true industrial shop applications such as machining, routing, and cutting — at a fraction of what it would cost to purchase one. Great for anyone who wants to automate a task in their home shop or small business. **$34.95**

SPECIAL OFFERS

The Amateur Scientist 3.0 The Complete Collection
by Bright Science, LLC



There are 1,000 projects on this **CD**, not to mention the additional technical info and bonus features. It doesn't matter if you're a complete novice looking to do their first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of your friends busy for a lifetime!
$26.95



The Getting Started Combo includes: Getting Started in Electronics by Author Forrest Mims and the DIY Electronics Kit. In his book, Mims teaches you the basics and takes you on a tour of analog and digital components. He explains how they work and shows you how they can be combined for various applications. The DIY Electronics Kit allows for the hands-on experience of putting circuits together — the kit has over 130 parts! No soldering is required and it includes its own 32 page illustrated manual.
Combo Price $62.95 Plus S/H

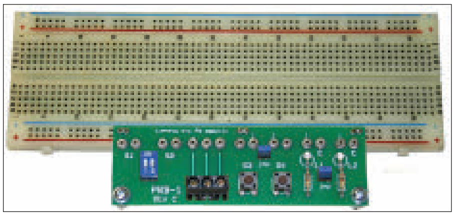
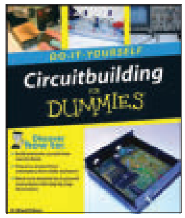
Or order online www.servomagazine.com

SPECIAL OFFERS

Proto Buddy Kit & Book Combo

For those just getting started in electronics as a hobby, a solderless breadboard (SBB) is the perfect platform for building those first circuits. Attach a Proto Buddy to an SBB, include a battery or two, and you will have a combo that has a lot of the same functionalities as more expensive units.

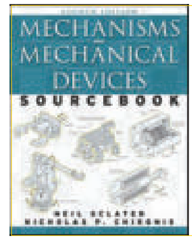
Combo includes PCB & Components, 830 point SBB, and *Do-It-Yourself Circuitbuilding For Dummies*.

Combo Price $57.95 Plus S/H
Limited time offer.

Mechanisms and Mechanical Devices Sourcebook,

by Neil Sclater, Nicholas Chironis

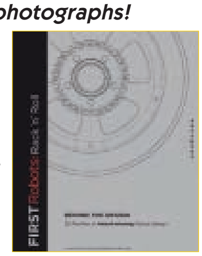


Over 2,000 drawings make this sourcebook a gold mine of information for learning and innovating in mechanical design. Overviews of robotics, rapid prototyping, MEMS, and nanotechnology will get you up-to-speed on these cutting-edge technologies. Easy-to-read tutorial chapters on the basics of mechanisms and motion control will introduce those subjects to you. **Reg $89.95 Sale Price $69.95**

FIRST Robots: Rack 'N' Roll: Behind the Design

by Vince Wilczynski, Stephanie Slezycki

More than 750 photographs!



The second annual book highlighting the creativity and process behind 30 winning robot designs from the 18th annual international FIRST Robotics Competition. The FIRST organization, founded by Dean Kamen (inventor of the Segway), promotes education in the sciences, technology, and engineering.

Reg $39.95 Sale Price $29.95

PROJECTS

Das Blinkenboard Kit

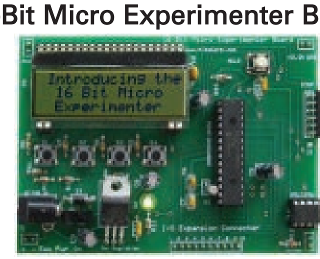
As seen in *Nuts & Volts* June issue, Personal Robotics by Vern Graner

This kit includes a pre-programmed ATtiny84 microcontroller that sports eight software PWM channels to control motor speed and light brightness. Jumper selectable patterns can be used to operate motors, solenoid valves, relays, or any DC load up to 24V/500 mA per channel!

Expand your board with the "Der Magnetfelder Detektor" component pack.

Subscriber's Price **$32.45**
Non-Subscriber's Price **$35.95**
PCBs can be bought separately.

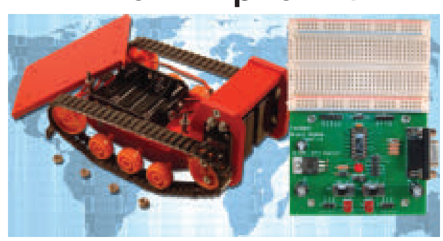
16-Bit Micro Experimenter Board



Ready to move on from eight-bit to 16-bit microcontrollers? Well, you're in luck! In the December 2009 *Nuts & Volts* issue, we will introduce you to the new, "the 16-bit Micro Experimenter." The kit comes with a CD-ROM that contains details on assembly, operation, as well as an assortment of ready-made applications. New applications will be added in coming months.

Subscriber's Price **$55.95**
Non-Subscriber's Price **$59.95**

Tankbot Kit & Brain Alpha Kit



As seen in the Sept. issue Tankbot/ Brain Alpha by Ron Hackett

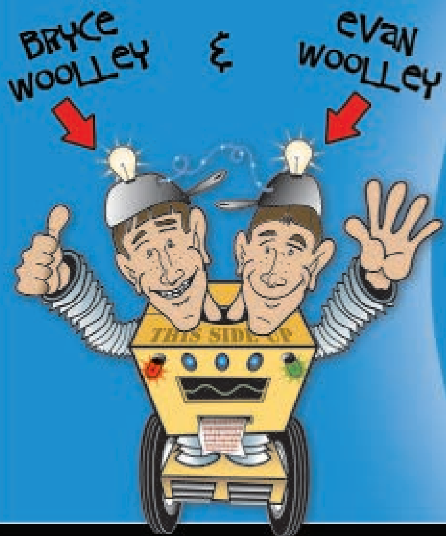
A series filled with projects and experiments to challenge you through your learning process while you grow your fully expandable Brain Alpha PCB!

The brain is a PICAXE-14A!

For more info & pictures, visit the *SERVO* Webstore.
Tankbot and the Brain Alpha Kit can be purchased separately.

Combo Price $ 138.95

Twin Tweaks



THIS MONTH:

Pavlov's Bots

THE UNDISCIPLINED ROBONI-I

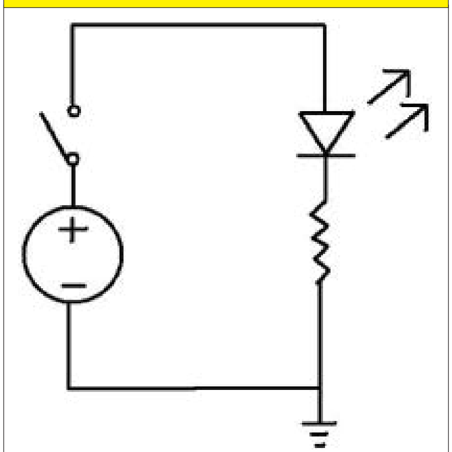
In November 2009, we had the privilege to introduce the Roboni-i from the ambitious folks at Robonica. The Roboni-i aims to revolutionize the electronics entertainment industry with its unique blend of a personality filled robot, structured game play, and an online world. While our playtime with the Roboni-i convinced us that the game of Space Pods is enough to keep even dedicated gamers busy for a long time, we are sure that a lot of *SERVO* readers are perhaps even more interested in the hacking capabilities of the lunar-roving bot. The shell of the Roboni-i includes a transparent blue panel that gives users a glimpse of its inner workings that should intrigue tinkerers and send them scrambling for a screwdriver.

Unconditioned Behavior

The Roboni-i comes with a compelling personality. If you leave it alone for too long it will go exploring on its own, with its responses to obstacles and adventurousness determined by how much you play with it. The Roboni-i is fun to drive around with its cantilevered wheels and a distinctive swaying gait. It is an acrobatic automaton, and it often flips over when you rapidly change directions. The flipping takes some getting used to, and it seemed like sometimes by just driving the bot around it was flipping out more than Jeff Lewis. We knew this was more a function of our poor driving than of the robot's rebelliousness, but it inspired us to pursue some behavior modification in the hopes of making us passable drivers.

We thought that pursuing some operant conditioning would be a fine way to alter the driving habits of the Roboni-i. The persistent flipping was the unconditioned behavior, and we thought that a punishment stimulus would help achieve our conditioned behavior of steady driving. B.F. Skinner might scoff at our methodology because we really want to condition the driver by punishing the robot, but with so much personality we think empathy with the Roboni-i is an understandable presupposition.

PROPOSED BEHAVIOR MODIFICATION CIRCUIT.



By punishment we didn't mean to do anything truly traumatic to the robot, but rather to elicit a response that would discourage the driver from testing the bot's acrobatic capabilities. We thought an added sensor that would activate upon the

Roboni-i's inversion would be a fine analog to the Skinner box, and we could program an unpleasant response to coincide with the inversion. We could program the Roboni-i to shiver upon activation of the sensor (kind of like a rat on Skinner's electrified floor).

So, what kind of device would we want to use? We played with the idea of a mercury switch, but that seemed ill-suited to a bot that targeted a younger demographic. A gravity switch would be a much safer way to go, and it would be easy enough to make from scratch.

Getting Our Bearings

Our plan was to use a ball bearing in a cylinder that would roll to the other end when the Roboni-i flipped over, thus completing a circuit between two wire ends. Technically, that would be all we needed for a bare bones circuit, but where's the fun in that? We wanted to include an LED in the circuit for cool factor as a debugging tool. As we constructed the circuit on the breadboard, the LED would be a convenient way to check that the switch was working, and it would also provide for a more multisensory reaction when Roboni-i engaged in the unconditioned behavior of flipping over. The circuit was still devilishly simple, of course, because in addition to the LED we would just need a pull-down resistor to eliminate floating voltages. We sketched out the circuit and began gathering the supplies.

We were hoping to find some ball bearings for sale at our local home improvement store, but our search was fruitless. Thankfully during the heyday of Robot Central, we had acquired a multitude of useful parts, and bearings were part of our collection. Liberating the balls from the bearing, however, would require some finesse.

Our preferred form of finesse was to use a radial cutting wheel, and we were sure to don our safety glasses before letting the sparks fly. We cut through the outer casing of the bearing and we took care to penetrate only the outer layer to avoid scuffing up the balls. Even after cutting the casing we had to use a screwdriver to pry the bearing apart, and we were cautious in our application of force to ensure that our fears of flying ball bearings were not realized.

We needed a cylinder to encase the ball bearing, and a straw happened to be the perfect size. It might have been too perfect, however. We wanted the ball bearing to complete a circuit when the robot inverted, so we had two wire ends poke through the sides of the straw. The tolerance was so tight though that even the slightest touch from one wire would stop the ball in its tracks. We needed the ball bearing to contact both wires, so we had to find a cylinder that had a slightly larger diameter.

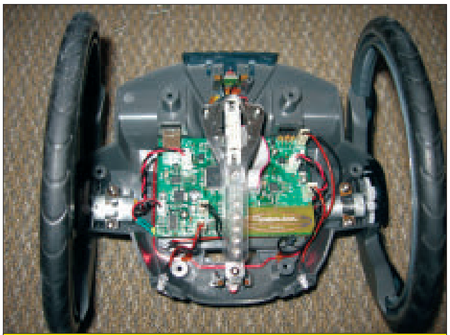
We noticed the cap on the pen that we used to draw our circuit diagram. A quick snip with the flush cuts eliminated the extra plastic, and we had a perfect cylinder already sealed on one end. Also useful was the fact that the more rigid plastic of the pen cap was much less prone to squishing and deformation than the flimsy straw.

Pigeonholed

To determine what kind of connector we needed to attach to our circuit, we needed to take a look inside the Roboni-i. We have to admit, we approached this task with a hint of trepidation. Many robot kits are designed to be hacked and are correspondingly easy to crack open. However, many bots that are targeted to a younger

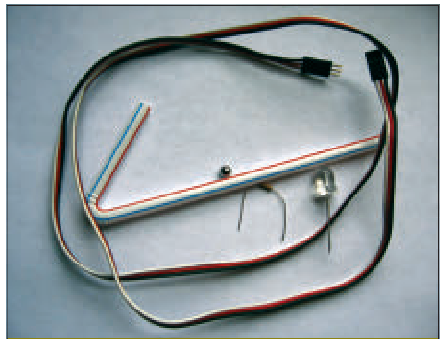
CUTTING OPEN THE BEARING.

RESULTS OF THE DISSECTION.



EXPLORATORY SURGERY.

IN THE BRAIN OF THE BEAST.



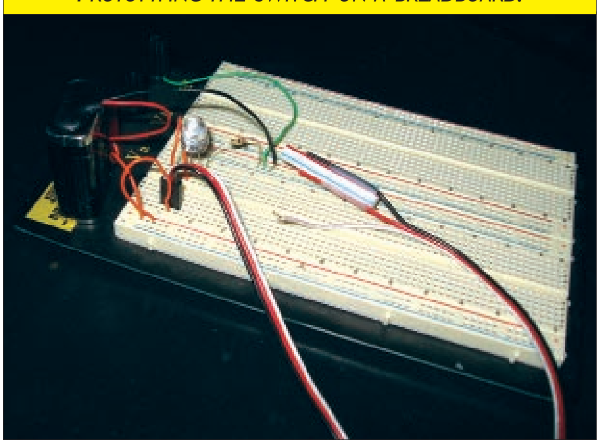
GRAVITY SWITCH MATERIALS.

demographic seem to deliberately interfere with hacking by making the casing difficult to remove. Screws are numerous and hidden. The casing might be difficult to pull off and fit back on. And sometimes once the casing finally does submit, the circuit board is an inaccessible rat's nest of wires that leave no open places to solder in your own devices.

Given the fact that the Roboni-i appears to be first and foremost an electronics entertainment system, we were afraid its hackability might be a limited afterthought. Thankfully, we were completely mistaken. The main body panel of the Roboni-i is held on with six screws that are easy to access. After removing the bot's head, the panel is easy to slide out from under the wheels. The panel isn't obnoxiously held on with wires or sensors inextricably bound to it with glue or stubborn connectors. Upon taking a look at the circuit board, we were excited to begin hacking. We were hoping to find some open pins or solder pads, and we were not disappointed.

The Roboni-i does indeed have a multitude of open solder pads; three clusters, in fact. One cluster is labeled JTAG, another JC, and another AC LED. We thought that at least one of these pads would provide the perfect way to incorporate our Skinner switch, but just to make sure we pulled out our trusty multimeter for a quick diagnosis. The JC and JTAG clusters gave us some small voltages, and the AC LED read about 3.7V. We were hoping a look into the programming interface might give us a few more clues about how to implement our switch, but we were confident that we would find a way to include our device.

PROTOTYPING THE SWITCH ON A BREADBOARD.



Before soldering up our gravity switch, we wanted to prototype it on a breadboard as a quick proof of concept. We were confident that our simple circuit design was sound, but we wanted to make sure that the gravity switch was consistent after our troubles with the straw. All we needed for the circuit was our cylinder, ball bearing, LED, 100K Ω pull-down resistor, and some extra wire. Anticipating a connection to the free solder pads, we used a male PWM cable for our connecting wire. When we first flipped over our cylinder, we were disappointed to see that the LED failed to light.

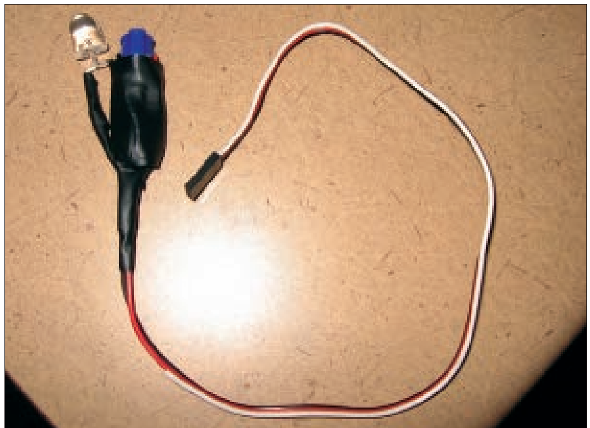
After inspecting the switch, we determined that one wire was catching the ball bearing before it could hit the second wire. After a little finesse with some needle-nose pliers, the switch was working like a charm. Each time we flipped it over, the LED lit up with a bright red and we were ready to solder it all up. Ideally, we would have liked to use some heat shrink to make everything look nice, but we were fresh out and electrical tape was a fine (albeit not as aesthetically pleasing) substitute. It looked like the Roboni-i had performed its last unpunished flip. All that was left to do was to hook it into the robot and program a traumatic reaction.

Like a Cat in a Puzzle Box

Our multimeter investigation was inconclusive, so we hoped that looking into the programming for the Roboni-i would give us some clues about how to implement our new switch.

The Roboni-i comes with a comprehensive software package known as the Command Center

SOLDERING UP THE GRAVITY SWITCH.



THE COMPLETED GRAVITY SWITCH.

which serves as a way to keep track of your Roboni-i's stats, a way to program your own games and behavior, and as a gateway into the online world. All of the different functions of the Command Center are easily navigated using a tabbed navbar. Last but certainly not least, the Roboni-i connects to the computer using a USB cable. A USB cable! We simply cannot express in words how thrilled we are that a company was courageous and forward-thinking enough to use a USB connection instead of an RS-232 serial connection (particularly because to the target laptop-toting video gamer demographic a serial port is a relic of the past ... something on the order of floppy disks and vacuum tubes). The Roboni-i kit even comes with the requisite Type B USB cable needed for downloading programs. Now that's classy!

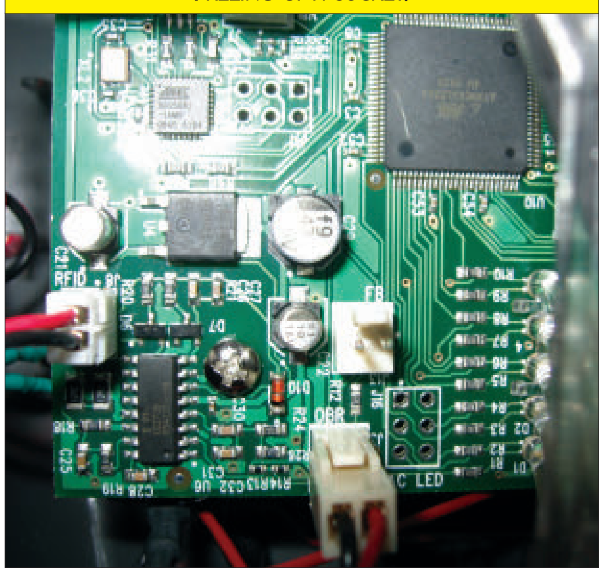
The Roboni-i Command Center features a graphical programming interface that essentially involves placing blocks onto a timeline. Contrary to what John Watson might say, the Roboni-i is not a tabula rasa. It comes with preprogrammed reactions to all of its sensor inputs. The user has the option of creating their own routines or simply modifying the bot's ingrained reactions. One screen of the Command Center shows a picture of the Roboni-i with arrows pointing to all of the sensors. The user only needs to click on the sensor they want to reprogram and they will be brought to the timeline editor.

The programming interface also includes numerous preprogrammed actions that help even complete programming neophytes to construct complex reactions to sensor inputs. There are premade blocks that make the robot shiver, recoil, and crab walk. The user can easily manipulate the duration of each action, and each block allows the user to complement the action with LEDs turning on or flashing. There is also a host of sounds.

When we first fired up the Command Center, however, we found that some of these amusing actions were unavailable to us because our Robot TX level was too low. Users can raise their Robot TX (robot trust) level by logging more hours playing with their Roboni-i. The Command Center tracks your time with the Roboni-i, and it documents autonomous time and driver controlled time in amusing pie charts. Line graphs track hours logged over time, and the whole setup looks very sophisticated.

Unlockable blocks aside, we unfortunately could see no clear way to include input from the free solder pad clusters. Perhaps we weren't looking deep enough, but the graphical interface seemed chiefly concerned with programming the Roboni-i's responses to its existing sensors. There were no free arrows on the starting screen and though we weren't expecting something as obvious as free arrows pointing to the "hacker ports," we were a bit stymied as to how to implement our newly made sensor.

FREEING UP A SOCKET.



READY FOR CONDITIONING.

Psychic Driving

However, the Roboni-i did present a fantastically easy way to incorporate new sensors. All of the existing sensors in the main body of the bot were connected to the PCB using socketed connectors (instead of being soldered directly into the board). An easy way to integrate our sensor would be to pirate one of the existing sockets. The front bumper seemed like an ideal candidate, for a quick and easy way to implement our new switch. We did have to trade in our male PWM connector for a female connector, but it was an easy switch.

Just as a knock on the front bumper completed the circuit by closing the SPST switch, our gravity switch would complete a circuit upon the execution of an acrobatic flip. Even without reprogramming the bot, we were able to see that the sensor worked because Roboni-i would back up and turn upon inversion (which seemed like an odd but observable reaction).

The Roboni-i is a stylish looking robot with a sleek outer shell that looks both cool and efficient. Unfortunately, the shell left no obvious opening for our PWM cable to escape, so we had to make one. We carved a small notch out of the transparent blue panel, and we were pleased to see that the final result was not atrociously noticeable.

With our surreptitious sensor switcheroo, all we had to do now was reprogram the reaction for the front bumper into something more appropriate for our operant conditioning. We liked the shiver function, but it was technically unavailable because our Robot TX level was still hovering around the amount of trust that John Watson's Little Albert had for white furry things. Fortunately, a short shiver was already a part of the front bumper reaction, so we kept that block and removed the other extraneous movements. We extended the time of the shiver to make sure it was very apparent to the errant driver. Even if the driver wasn't very sympathetic to the inverted bot, a long shiver would reduce driving ability enough to motivate a change in behavior. We also made sure the LEDs were flashing and that the Roboni-i would make an "irritated" sound so the driver knew that learning how to drive carefully was a worthwhile endeavor.

We were really conditioning the driver and not the robot, so we were careful to make our modification only to the driving mode. Users can also modify Roboni-i's autonomous behaviors, but it was nice to know that we wouldn't cause any permanent psychological damage.

After downloading the program (downloading programs to the Roboni-i is a snap), we were ready to test the efficacy of our behavior modification technique. It takes just a quick change of direction to send the Roboni-i into an acrobatic loop, and we were pleased to see the robot lapse into a prolonged shiver after flipping over. Other drivers might not be as sympathetic, but we sure wanted to drive more steadily to spare the Roboni-i from punishment.

Now, it may have been easier just to drive more slowly in the first place, but we were thrilled to discover that the Roboni-i is indeed readily hackable. The sockets on the circuit board allow users to immediately attach custom sensors and program them without any sophisticated tricks. The timeline programming interface is intuitive and easy to use, but with a little creativity complicated actions can be programmed in. We don't think that the Robot TX level would present a serious obstacle to any new

PROGRAMMING THE ROBONI-I.

RECOMMENDED WEBSITES

www.robonica.com
www.roboni-i.com

Roboni-i user, especially if they take the time to drive normally while learning preloaded games (like Colors).

That's a Hack

The folks at Robonica are committed to making the bot an entertaining platform for all robot tinkering demographics. In the online world, feedback and suggestions are welcomed from players, and the online world is evolving into something very exciting.

That might sound like great news for just the gamers, but there's also great news for the hackers! Robonica is coming out with a Software Development Kit, so if finessing the graphical timeline is below your coding skills, fear not! The Software Development Kit will include an Application Programming Interface that should keep even the most experienced hackers entertained.

Overall, we were thrilled that the Roboni-i presents the rare combination of being an excellent toy for kids and an accessible hacking platform for tinkerers. This kind of robot kit is most exciting because it has the potential for someone mainly interested in online gaming becoming interested in science and engineering. With the easily opened shell and the ability to integrate new sensors by stealing existing sockets, the Roboni-i invites curiosity and experimentation. This is great because one of the biggest obstacles to getting kids interested in science and engineering is that it's intimidating. To the uninitiated, breadboards and circuit diagrams can be overwhelming. But when experimenting with linear circuits is really just a way to give your game-playing robot a competitive edge, it all becomes a part of the game. Couple this with Robonica's admirable commitment to acting on user feedback, everyone has the chance to change the Roboni-i's behavior without the help operant conditioning. **SV**

GETTING A SHOCK!

More Good News For Hackers

Physically hacking into the Roboni-i is refreshingly simple – the easily removed casing, the free solder pad clusters, and the socketed sensors make it smooth sailing to attach custom sensors. Our only difficulty was interfacing our sensor with the programming, but the folks at Robonica have anticipated that intrepid tinkerers will be looking for a more elegant way to modify the bot's behavior with code as opposed to operant conditioning.

A Software Development Kit (SDK) will be available for users that want to have more control to create their own games for the Roboni-i or to be able to modify any of its external functions. The Application Programming Interface (API) uses standard C with open-source compilers which should be a welcome sight for hackers more at home with *for* loops than blocks on a timeline.

Courageous coders can manipulate every facet of how the Roboni-i interacts with the outside world. Its movement, interaction with base stations, and interaction with the RFID tags in the SFX and other cards can all be reprogrammed. Users can even change how the Roboni-i uses Zigbee messages to communicate with remote controls and other robots.

The custom code created using the API can be downloaded into one of the Roboni-i's three memory slots used for games. This is good news for cautious programmers that don't want to risk the whole of their robot's behavior on an untested program. Even if things went awry, at least the other two game slots could remain untouched. The SDK also has options for gung ho programmers on the other side of the spectrum of cautiousness. New firmware can be downloaded to the Roboni-i, its personality code can be modified, and even the operating system is not immune to modification.

For those chomping at the bit for more technical details, the main microcontroller for the Roboni-i is an Atmel ATXMEGA128A1MCU. The Roboni-i also employs microcontrollers for interaction with the IR Pod and base station and the RF communications. For Zigbee communications, the Roboni-i uses an AT86F230 RF transceiver. For hackers looking to realize the full potential of the Roboni-i's powerful brain, the SDK offers an excellent way to modify the robot's behavior beyond what can be achieved using the default graphical programming.

The SDK is just another indicator of Robonica's commitment to providing a platform that is as compelling as a platform for experimentation as it is a video game. The SDK will make integration and implementation of custom sensors a snap using familiar C programming, and it can all be done without the electric trauma of a Skinner Box.

Then and NOW

A CLOSER LOOK AT PERSONAL SERVICE ROBOTS

by Tom Carroll

Last month, I wrote about service robots and how they've become the largest segment of the robotics industry, by far. This is not to say that the industrial robot category has not changed, as these robots have become far more capable with active vision systems, precision direct-drive motor systems, and cost less to implement. This recession has hit all industries very hard and the robotics industry as a whole has felt the brunt of it, especially the service robot segment. It is hard enough for factory management to approve capital funding for new assembly line robots here in the US; the general public simply cannot afford to buy the service robots that have been developed in the last few years, such as Waseda University's versatile Twendy-one shown in ***Figure 1****. Weighing in at 250 pounds, this meter and a half tall robot is expected to be commercially produced by 2015.*

A few years back, about the only thing that a typical personal robot could do was to wander around the house, act as a pet, or maybe remind elderly folks to take their medications or possibly make a call for them.

Personal robots today are a lot more sophisticated with functional appendages that can assist a person or play a game with them. This is the segment of service robots that are the most interesting — a mobile robot that is designed to work or exist among humans. Today's personal service robots can interact with people in numerous ways.

In my July '06 *SERVO* article, I wrote about robots who care for people. In that article, I centered on robots that operated within the confines of a person's home to serve and assist elderly or disabled people. Robots can

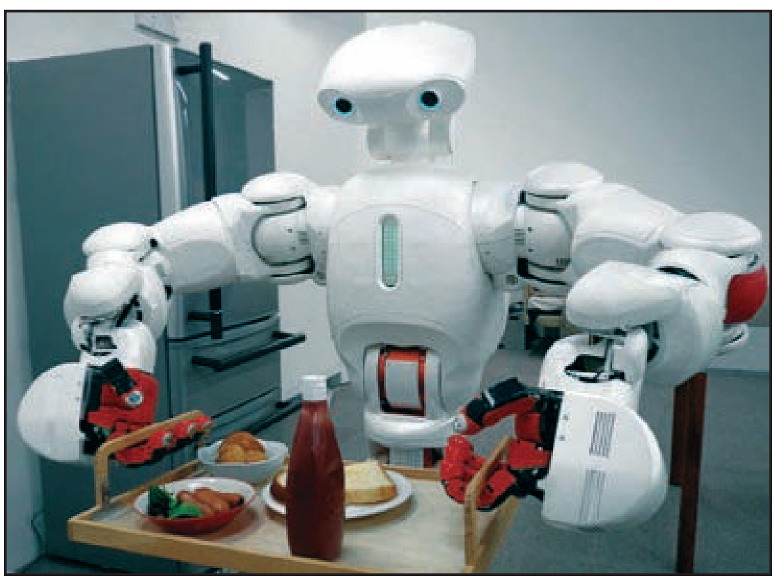


FIGURE 1. Twendy-one serving a meal.

personally interact and care for humans in many other ways, including assisting in hospitals, in emergency situations, as entertainers and even in the workplace along side their "industrialized" cousins. They can be used as a simple animal robot to lie in the lap of an elderly person and comfort them, such as the $6,000 Paro baby harp seal robot shown in **Figure 2** that recently received FDA approval here in the US. These robots can act as guides in museums or other unfamiliar places such as assistants to astronauts in deep space. Let's take a look at a few entirely different types of service bots that have made headlines lately.

Personal Service Robot Designs

Personal service robots that work among humans can have wheels, tracks, legs, or even fly or swim. Most of today's personal robots still do not have any appendages, but that does not make them any less useful for the tasks they are designed for. Automated Guided Vehicles (AGVs) have been used in factories and hospitals for years to move everything from medicines to factory materials and finished products, moving silently around people. These robots rarely interact with humans directly other than stopping when someone crosses its path or waiting patiently as a person loads or unloads its pallet of materials. These basic personal robots have cut operating costs in factories, warehouses, offices, and hospitals around the world. The Pyxis HelpMate robot shown in **Figure 3** is a good example of an AGV used in hospitals to transfer materials. It was first designed by Joe Engelberger's company in the '80s, that was later bought out by Pyxis.

Emergency Services Robots

War, extreme weather and other natural disasters, man-made disasters, terrorism, and political unrest are making the headlines every day. With each of these strikes against mankind, rubble and destruction are the result. People often are trapped under many tons of debris.

Military and law enforcement robots have been used in search and rescue operations, but a more specialized type of robot is better suited for use at these disaster scenes. With uneven ground, possible fires, and other hazardous situations, plus obstacles and very small areas in which to traverse, traditional robots are not well-suited for this.

FIGURE 2.Christine Hsu of Paro Robots US holding Paro, the baby harp seal.

Dr. Robin Murphy's Rescue Robots

Enter the Inuktun Extreme robot developed by Dr. Robin Murphy. With a multi-media communications 'head' attached, it can act as a 'survivor buddy' to a person trapped under rubble. In many cases where people have been trapped, (as in a mine collapse), the use of the Inuktun robot shown in **Figure 4** to confirm survivor

FIGURE 3. The Pyxis robot used in hospitals.

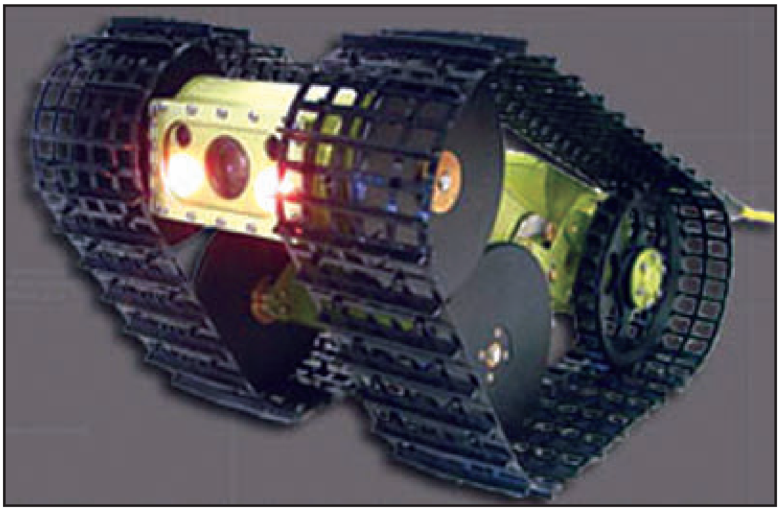


FIGURE 4. Inuktun Extreme.

FIGURE 5. Dr. Robin Murphy with Inuktun.

numbers and casualties, and to provide lighting, water, reassuring voices, and even soothing music until rescue is priceless. Dr. Murphy was first turned onto rescue robots when she was a professor of Computer Science and Engineering at the University of South Florida. She viewed the destruction caused by the Oklahoma City bombing and saw the difficulty rescuers had crawling into tight spaces within the rubble. She had received a B.S. in Mechanical Engineering and her M.S. and Ph.D. in Computer Science at Georgia Tech, which was the ideal educational background for robot design.

When 9-11 attacks occurred on the World Trade Center, she knew that her robots could go deeper and further than human rescuers and was on the scene within 24 hours for their first official mission. Other rescuers were using trained dogs and TV cameras mounted on poles, and quickly saw the unique advantages of the Inuktuns that could go 60 feet through smoldering rubble. Her designs were later incorporated into more autonomous rescue robots that were built by American Standard Robotics and Remotec — a subsidiary of Northrop Grumman. She shared a Microsoft grant with a Stanford professor to develop and refine her survivor buddy concept.

Other world-wide disasters such as Hurricane Katrina, the damage caused by the Southeast Asian Tsunami of 2005, the Brazilian mudslide, and the more recent devastating earthquake in Haiti have shown a clear need for these specialized robots. Newer versions could not only communicate with potential victims to be rescued, but deliver water, gas masks, oxygen, bandages, medications, and food. Larger versions could be used on the battlefield for these same functions, but also shield soldiers under fire and retrieve wounded ones.

Now a professor at Texas A&M, Dr. Murphy is developing robots, and preparing and honing them in a very unique facility. The 52-acre Disaster City located in nearby College Station, TX, is one wreck of a place. **Figure 5** shows a photo taken at Disaster City by Jeff Wilson of *Wired Magazine*. "Created by the Texas Engineering Extension Service (TEEX) — a member of the Texas A&M system — the mock community features full-scale, collapsible structures

FIGURE 6. The robot doctor — nurse developed in New Zealand.

FIGURE 7. Charlie with HealthBots Rebecca, Bruce, and Elizabeth, Univ. of Auckland.

is right next to a fire fighting training center. Personnel from around the world who have attended customized rescue scenarios say Disaster City makes the sets at Universal Studios Hollywood look downright tame.

"While rescue robots aren't perfect, they are "good enough" to be used. I see the future of rescue robotics as the confluence of two streams: One is about continuing to improve the technology, the other making the technology available. To paraphrase the old saying, a rescue robot in the responder's hands immediately is worth two in a lab around the world," commented Dr. Murphy.

designed to simulate various levels of disaster and wreckage which can be customized for the specific training needs of any group." A carefully-prepared hodge-podge of wrecked buildings, rubble and anything else one can imagine is used to train rescue personnel, as well as test rescue robots. It

A Robot Nurse and Medical Doctor?

It appears that researchers in Auckland, New Zealand are taking autonomous medical robotics a bit past the Da Vinci surgical robots. The balancing

FIGURE 8. Murata — robot nurse.

FIGURE 9. CMU's SnackBot.

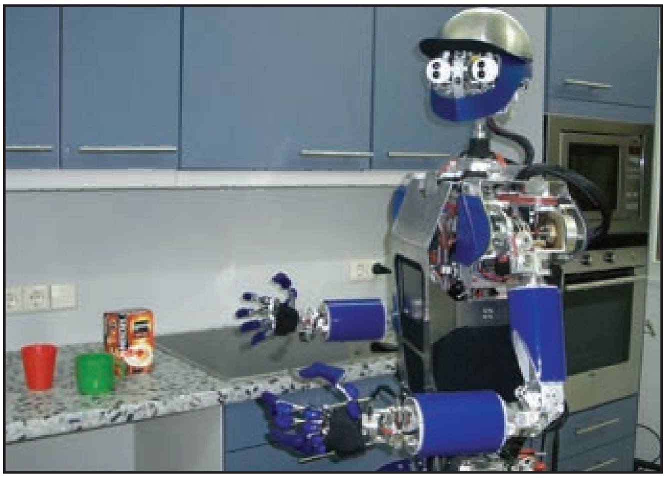


FIGURE 10. Anthropomorphic robot Armar III.

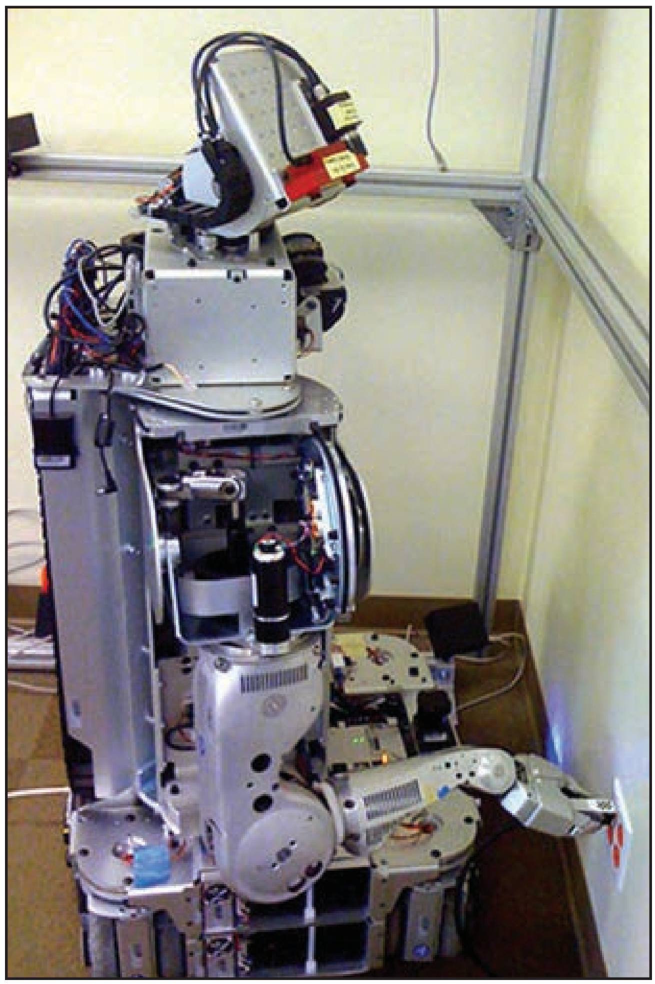
FIGURE 11. Completed grippers for the PR-2.

robot in **Figure 6** appears to be using a standard stethoscope on a patient. The NZ researchers have used over $5 million in grants working along with Korean researchers to develop a robot that can take blood pressure, heart beat, and similar measurements.

Rather than emulating an actual doctor, the robot will circle a waiting room, take vitals from patients, and reassure them that a real doctor will be with them soon. The robot is the first in a series of proof-of-concepts to lead the researchers to an automated health system. The real turning point will be when humans accept robots in place of real doctors. The Selwyn Retirement Village in Auckland uses the robot nicknamed Charlie (shown in **Figure 7**) to greet people via the touch screen, take the residents blood pressure, and deliver medications as required. Similar to the RP-7 by InTouch Health Systems, the robot is built by Yujin Robot of Korea for HealthBots. The three foot, six inch, 99 pound Charlie can also call for assistance should someone fall.

Murata Machinery in cooperation with NEDO (a Japanese advisory group) has developed the robotic nurse shown in **Figure 8.** This robot nurse was developed as part of Japan's "Project for Strategic Development of Advanced Robotics Elemental Technologies" led by NEDO. It is touted by the media to deliver medications to patients in the hospital, especially during late night hours. This doesn't mean that it will enter a patient's room and jab them with a needle, but rather it will enter and alert the patient to take their medication from a tray and swallow them with a provided drink of water. Again, like the New Zealand research robot, this is a first step in truly implementing robots in the hospital environment, (this is similar to the earlier HelpMate robot built by Joe Engelberger's company). In Japan, nurses earn a higher salary than they do here in the US, so robotizing some of the more mundane tasks is economic, plus it will allow nurses to concentrate on more complex medical procedures for which they were trained.

FIGURE 12. The PR-2 from Willow Garage.



Carnegie Mellon's NurseBot, Pearl

The US certainly has not been lax in advanced medical robot research. Aside from the high-level surgical systems, personal patient care by robots has been investigated by several universities. Initial thoughts of potential medical robot research projects always look at autonomous, higher-level medical intervention first, but practical designs with less legal implications always come into play. Injections of medicines, and internal and external probing — even with a stethoscope — require a trained ear and eye to make an accurate diagnosis. For this reason, teams from the different universities steered designs away from more complex nurse designs and towards benign robots that verbally interact with patients, offering them snacks and souvenirs to be taken from a tray. The Carnegie Mellon University's SnackBot shown in **Figure 9** is the latest version of the original 2001 CMU Pearl NurseBot concept. Pearl relied on a touch screen command system whereas SnackBot relies on verbal communications. SnackBot has a SICK laser range-finding system for navigation with face recognition to distinguish different people and is used primarily as a human-robot interaction study model.

"SnackBot is a mobile autonomous robot intended for both fully autonomous and semi-autonomous operation, built by an interdisciplinary team at Carnegie Mellon University. SnackBot has two jobs. One job is to serve as a research platform for projects in robotics, design, and behavioral science. We welcome new partners or sponsors for this work," said CMU. "SnackBot's other job is to serve snacks."

FIGURE 13. PR-2 conceptual drawings.

Robots in the Home

Here are a couple personal service robots that just might make it to our homes soon. Armar (shown in **Figure 10**), created by the Institute of Product Development at the University of Karlsruhe in Germany, was created as a machine that closely cooperates with humans. With stereo cameras in its head, as well as hands, the robot has been mostly "trained" for kitchen tasks.

The PR-2 from Willow Garage

Willow Garage — located in Menlo Park, CA — is anything but a 'garage'. It has produced one of the most capable and beautifully-designed personal robots as demonstrated by the rows of beautifully-machined grippers you can see in **Figure 11**. **Figure 12** shows some of the interior components of a prototype, with the caster base and two anthropomorphic arms. Steve Cousins, CEO, is quite proud of their designs and anticipates 10 qualified researchers to be using PR-2 beta models (shown in the conceptual drawings in **Figure 13**) for further research. Interchangeability and upgradability of various systems, three to six hours of operating time on a 1.3 kWh battery power system, four full Intel Core Duo 2 computers, and two 7DOF arms round out a few of the key systems of the PR-2.

Safety was a key factor in the robot's design as all large robots working in close proximity to humans have some degree of inherent danger if used incorrectly. As Cousins states, "The safe use of a robot in close proximity to people is a difficult problem, and is an active area of development in the robotics community. By developing a robot that is safe enough to interact physically and with the level of PR2, we hope to support and advance research development in the area of true robot safety." Go to **willowgarage.com** for more information.

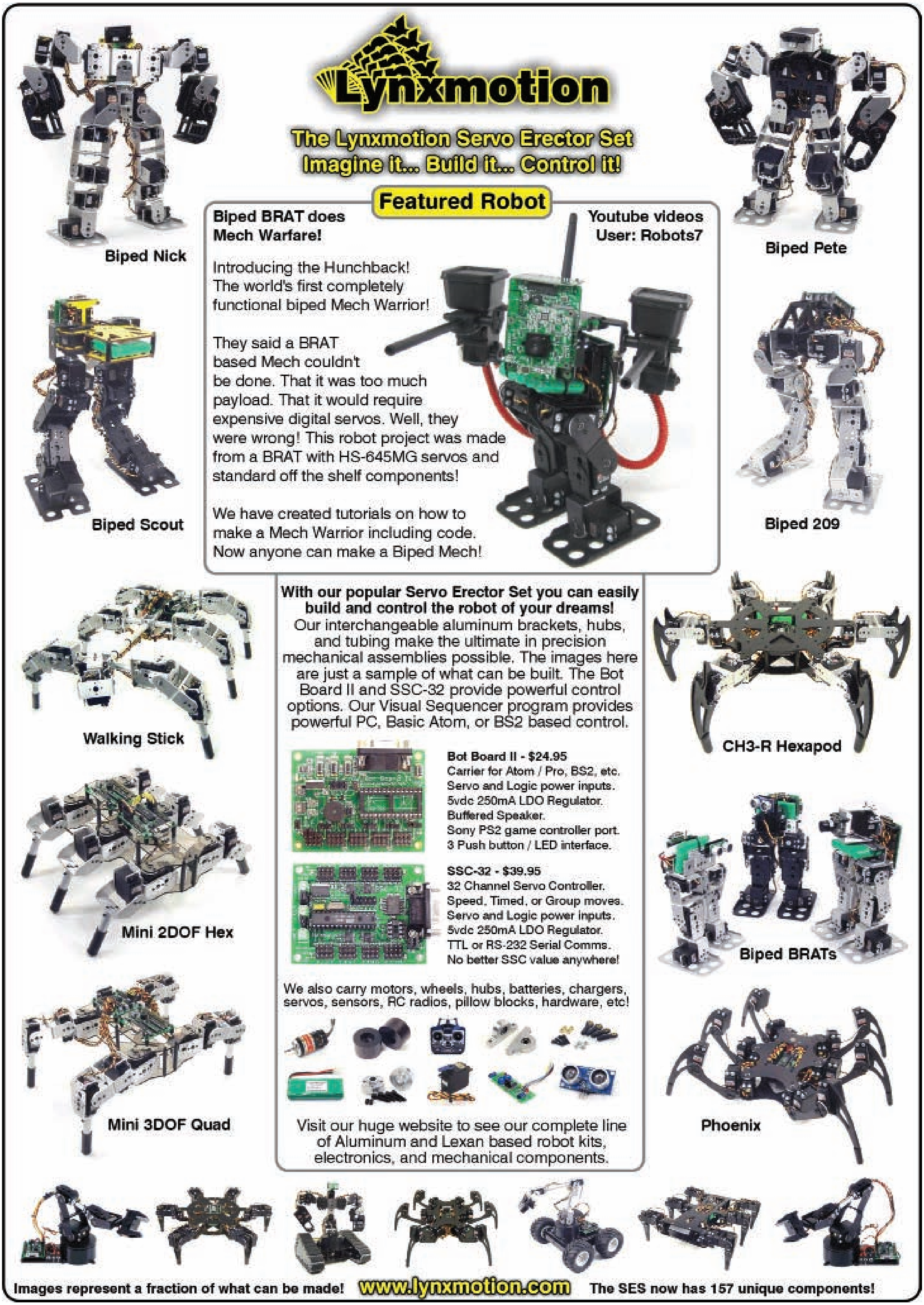
Final Thoughts

Home experimenters are producing cutting-edge robots in garage workshops. Universities and companies around the world sense the great need to create robots that work and serve alongside people in their daily lives. The new decade is sure to bring some amazing progress in personal service robots. **SV**

Tom Carroll can be reached at TWCarroll@aol.com.

ROBO-LINKS

Advertiser Index



Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Biped Nick
Biped Pete
Featured Robot
Biped BRAT does Mech Warfare!
Youtube videos User: Robots7
Introducing the Hunchback! The world's first completely functional biped Mech Warrior!
They said a BRAT based Mech couldn't be done. That it was too much payload. That it would require expensive digital servos. Well, they were wrong! This robot project was made from a BRAT with HS-645MG servos and standard off the shelf components!
We have created tutorials on how to make a Mech Warrior including code. Now anyone can make a Biped Mech!
Biped Scout
Biped 209
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible. The images here are just a sample of what can be built. The Bot Board II and SSC-32 provide powerful control options. Our Visual Sequencer program provides powerful PC, Basic Atom, or BS2 based control.
Walking Stick
CH3-R Hexapod
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!
Mini 2DOF Hex
Biped BRATs
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Mini 3DOF Quad
Phoenix
Visit our huge website to see our complete line of Aluminum and Lexan based robot kits, electronics, and mechanical components.
Images represent a fraction of what can be made!
www.lynxmotion.com
The SES now has 157 unique components!